Segmentation by deep learning with geometric constraints and active contours.

Nicolas Makaroff

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'Université Paris-Dauphine PSL

Segmentation by Deep Learning with Geometric Constraints and Active Contours.

Soutenue par
Nicolas Makaroff
Le 10 décembre 2024

École doctorale n°543
SDOSE

Spécialité
Mathématiques Appliquées

Préparée au
CEREMADE

Composition du jury :

Carole LE GUYADER INSA Rouen Normandie	*Présidente du jury*
Abderrahim ELMOATAZ Université de Caen Basse Normandie	*Rapporteur*
Weihong GUO Case Western Reserve University	*Rapporteure*
Da CHEN Shandong Artificial Intelligence Institute	*Examinateur*
Jean-Marie MIREBEAU ENS Paris-Saclay	*Examinateur*
Laurent D. COHEN CEREMADE	*Directeur de thèse*

Dauphine | PSL
UNIVERSITÉ PARIS

Dauphine | PSL
UNIVERSITÉ PARIS

University Paris-Dauphine

Doctoral School **École Doctorale Sciences de la Décision, des Organisations, de la Société et de l'Échange**

University Department **Centre De Recherche en Mathématiques de la Décision, Université Paris-Dauphine**

Thesis defended by **Nicolas Makaroff**

Defended on **December 10, 2024**

To become Doctor from University Paris-Dauphine

Academic Field **Sciences**
Speciality **Mathématiques appliquées**

Segmentation by Deep Learning with Geometric Constraints and Active Contours.

Thesis supervised by *Laurent D. COHEN*
Committee members

President of the Jury	Carole Le Guyader	Professor at INSA Rouen
Referees	Weihong Guo	Professor and chair at Case Western Reserve University
	Abderrahim Elmoataz	Professor at Université Caen Basse Normandie
Examiners	Jean Marie Mirebeau	Senior Researcher at ENS Paris-Saclay
	Da Chen	Associate Professor at Shandong Artificial Intelligence Institute
Supervisor	Laurent D. COHEN	Senior Researcher at l'University Paris Dauphine-PSL

COLOPHON

The doctoral dissertation entitled “Segmentation by Deep Learning with Geometric Constraints and Active Contours.”, written by Nicolas Makaroff, was completed on December 12, 2024, typeset with the document preparation system LaTeXand the yathesis class dedicated to theses prepared in France.

Dauphine | PSL
UNIVERSITÉ PARIS

Université Paris-Dauphine

École doctorale **École Doctorale Sciences de la Décision, des Organisations, de la Société et de l'Échange**

Unité de recherche **Centre De Recherche en Mathématiques de la Décision, Université Paris-Dauphine**

Thèse présentée par **Nicolas Makaroff**

Soutenue le **10 décembre 2024**

En vue de l'obtention du grade de docteur de l'Université Paris-Dauphine

Discipline **Sciences**
Spécialité **Mathématiques appliquées**

Segmentation by Deep Learning with Geometric Constraints and Active Contours.

Thèse dirigée par *Laurent D. COHEN*
Composition du jury

Présidente du Jury	Carole Le Guyader	Professeure à l'INSA Rouen
Rapporteurs	Weihong Guo	Professeure et chair à Case Western Reserve University
	Abderrahim Elmoataz	Professeur à l'Université Caen Basse Normandie
Examinateurs	Jean Marie Mirebeau	Directeur de recherche à l'ENS Paris-Saclay
	Da Chen	Associate Professeur à Shandong Artificial Intelligence Institute
Directeur de thèse	Laurent D. COHEN	Directeur de recherche à l'Université Paris Dauphine-PSL

This thesis has been prepared at

Centre De Recherche en Mathématiques de la Décision, Université Paris-Dauphine
Place du Maréchal De Lattre De Tassigny
75016 Paris
France

☎+33 1 44 27 42 98
Web Site: `http://ceremade.dauphine.fr`

Segmentation by Deep Learning with Geometric Constraints and Active Contours.

Abstract

Segmentation of medical images is crucial in clinical practice, requiring accurate and reliable methods to aid diagnosis and treatment planning. However, existing deep learning approaches often need more interpretability and robustness, limiting their application in sensitive clinical environments. This thesis addresses these challenges by proposing two new deep learning models integrating classical image processing techniques to improve segmentation performance and reliability.
The first contribution, the Chan-Vese Attention U-Net, incorporates an attention mechanism based on Chan-Vese energy minimisation into the U-Net architecture. This approach exploits geometric constraints to guide the segmentation process, enabling the model to produce more accurate and easier-to-interpret results by focusing on relevant regions of the image and minimising irrelevant details. The second contribution, Fast Marching Energy CNN, combines neural networks with geodesic distance computation to learn isotropic Riemannian metrics directly from the data, generating robust segmentation masks that preserve geometric and topological properties. These methods integrate differentiable distance transforms and the subgradient walk algorithm into a differentiable framework.
By integrating traditional energy minimisation techniques with modern deep learning models, this research advances the field of medical image analysis, providing more reliable and interpretable tools for automated segmentation. The results of this thesis can potentially improve clinical decision-making processes and the adoption of AI-driven solutions in healthcare.

Keywords: Deep Learning, Computer Vision, Attention Mechanism, Medical Data, Active Contours,

Geodesic Distances

Segmentation by Deep Learning with Geometric Constraints and Active Contours.

Résumé

La segmentation des images médicales est une tâche critique dans la pratique clinique, nécessitant des méthodes précises et fiables pour aider au diagnostic et à la planification du traitement. Cependant, les approches d'apprentissage profond existantes manquent souvent d'interprétabilité et de robustesse, ce qui limite leur application dans des environnements cliniques sensibles. Cette thèse aborde ces défis en proposant deux nouveaux modèles d'apprentissage profond qui intègrent des techniques classiques de traitement d'images pour améliorer la performance et la fiabilité de la segmentation.
La première contribution, le Chan-Vese Attention U-Net, incorpore un mécanisme d'attention basé sur la minimisation de l'énergie de Chan-Vese dans l'architecture U-Net. Cette approche exploite les contraintes géométriques pour guider le processus de segmentation, ce qui permet au modèle de produire des résultats plus précis et plus faciles à interpréter en se concentrant sur les régions pertinentes de l'image et en minimisant les détails non pertinents. La seconde contribution, le Fast Marching Energy CNN, combine les réseaux neuronaux avec le calcul de la distance géodésique pour apprendre les métriques riemanniennes isotropes directement à partir des données, ce qui permet de générer des masques de segmentation robustes qui conservent à la fois les propriétés géométriques et topologiques. Ces méthodes utilisent des transformées de distance différentiables et l'algorithme de marche sous-gradient pour les intégrer dans un cadre différentiables.
En intégrant les techniques traditionnelles de minimisation de l'énergie aux modèles modernes d'apprentissage profond, cette recherche fait progresser le domaine de l'analyse d'images médicales, en offrant des outils plus fiables et interprétables pour la segmentation automatisée. Les résultats de cette thèse ont le potentiel d'améliorer les processus de prise de décision clinique et l'adoption de solutions pilotées par l'IA dans les soins de santé.

Mots Clés: Apprentissage Profond, Vision par Ordinateur, Méchanisme d'Attention, Données Médicales, Contours Actifs, Distances Géodésiques

Remerciements

La rédaction de cette thèse fut longue, et je peux maintenant prendre le temps de remercier les personnes qui m'ont aidé de près ou de loin.

Tout d'abord, j'aimerais profondément remercier Weihong Guo et Abderrahim Elmoataz d'avoir accepté d'être rapporteur de ce manuscrit et d'avoir rendu possible la soutenance. J'aimerais aussi remercier les membres du jury Carole Le Guyader, Da Chen, Jean-Marie Mirebeau, qui ont accepté de faire partie du jury de ma thèse. C'est un grand honneur qu'ils me font et je leur en suis très reconnaissant.

Merci également à mon directeur de thèse, Laurent Cohen, pour son encadrement scientifique, ses conseils et son soutien matériel pour mener à bien mes recherches.

Un grand merci aux personnels du secrétariat et de l'informatique qui sont toujours là pour nous aider : Isabelle, Anne-Laure, César, Marko, Thomas et Gilles.

Merci à Emeric pour ses conseils et encouragements.

Merci à Sergio Pulido pour son soutien et ses conseils tout au long de mes études depuis mes années d'école d'ingénieur.

Je tiens à remercier chaleureusement Raphaël, avec qui j'ai partagé tant de temps durant cette thèse. Nos cafés matinaux et nos déjeuners anticipés à 11h40 pour éviter la foule ont rythmé mes journées de travail. Les moments passés ensemble à savourer des pintes ou simplement à profiter de moments de détente, ont été indispensables pour me changer les idées. Ta compagnie m'a offert le soutien et l'évasion dont j'avais besoin pendant ces années intenses. J'espère sincèrement que nous aurons l'occasion de continuer à nous retrouver et à créer de nouveaux souvenirs. Merci pour ton amitié solide et ton soutien constant.

Merci à Clément, Tom, Pierre et Mathieu d'avoir été présents tout au long de cette thèse. Les innombrables pintes partagées, les Grands Prix de F1 suivis avec passion, les matchs de rugby endiablés et tous les moments de détente passés ensemble ont été de véritables bouffées d'air frais. Les repas à l'appartement avec Tom et Clément, où nous refaisions le monde, comptent parmi mes meilleurs souvenirs. Votre amitié m'a permis de décompresser et de me ressourcer lorsque j'en avais le plus besoin. Je suis vraiment heureux de vous avoir dans ma vie et je vous remercie pour votre soutien indéfectible.

Merci à Rémi pour tous les moments passés attablés à Tours ou à Lille autour d'une bière depuis le lycée.

Merci à Elise d'avoir accepté à chaque fois de rentrer à pieds alors qu'elle était à vélo, après nos pintes place de Châteauneuf.

Merci à Chris et Gilles pour toute leur affection. Merci d'avoir toujours pris des nouvelles de l'avancement de ma thèse.

Merci à mes grands-parents pour tout leur amour et leur soutien. Merci de vous être oc-

cupés de moi lorsque j'étais petit, de m'avoir entouré de votre affection et de votre bienveillance. Votre intérêt constant pour mes études et vos encouragements m'ont toujours motivé à donner le meilleur de moi-même. Les moments précieux que nous avons partagés ont profondément marqué ma vie.

Merci à mes parents. Vous m'avez toujours soutenu dans mes choix, m'aidant à réaliser tout ce que je souhaitais entreprendre. Votre présence constante et votre encouragement m'ont permis d'arriver là où je suis aujourd'hui. Je suis fier d'être votre fils et vous suis profondément reconnaissant pour tout ce que vous avez fait pour moi.

Merci à mes frères, Thomas et Valentin. Thomas, notre compétition amicale dans le sport et les études jusqu'à la fin du lycée m'a toujours poussé à me dépasser. Ces défis partagés ont été une source de motivation et de nombreux souvenirs mémorables. Valentin, nos nombreuses heures de discussions sur les nouveaux jeux vidéo et le temps passé à jouer ensemble ont été des moments précieux. Merci de m'avoir fait découvrir des jeux comme The Last of Us, A Plague Tale et The Witcher. Les centaines d'heures passées tous les trois à détruire des orcs, et fuir devant Tharzog sur La Guerre du Nord resteront gravées dans ma mémoire.

Merci à Pops pour sa patience entre deux pâtés et d'accepter tous ces câlins et papouilles.

Merci enfin à Alexandra. Ta présence au cours de ces trois années a été essentielle. Tu as toujours été là pour m'écouter, me soutenir et m'encourager, même lorsque mes préoccupations pouvaient sembler abstraites ou complexes. Ta patience et ta compréhension m'ont aidé à traverser les moments difficiles, et ta joie de vivre a rendu les bons moments encore meilleurs. Tu es devenue le pilier sur lequel je peux toujours compter, et je suis profondément reconnaissant de t'avoir à mes côtés. Merci pour tout ce que tu fais et pour la personne que tu es.

Contents

List of Figures

List of Tables

Acronymes et anglicismes

CDT *Convolutional Distance Transform*

CNN *Convolutional Neural Networks*

CV *Computer Vision*

DL *Deep Learning*

GVF *Gradient Vector Flow*

LSTM *Long Short-Term Memory*

NLP *Natural Language Processing*

NN *Neural Network*

RNN *Recurrent Neural Networks*

ViT *Vision Transformers*

Résumé en Français

Les travaux présentés dans cette thèse ont pour objectif de proposer de nouvelles méthodes pour l'analyse d'images médicales, en particulier la segmentation des tumeurs cérébrales et des structures arborisées telles que les réseaux vasculaires. La structure de cette thèse est conçue de manière à permettre la compréhension des enjeux et des méthodes de l'état de l'art à partir de ce seul manuscrit, et avec les connaissances d'un étudiant de Master 2 en mathématiques appliquées, en informatique ou en apprentissage automatique.

Ces travaux s'inscrivent dans le cadre des méthodes récentes d'apprentissage automatique, plus précisément des méthodes d'apprentissage profond (Deep Learning), qui reposent sur l'entraînement de grands réseaux de neurones pour réaliser des inférences. Nous testons également des méthodes plus anciennes, telles que les contours actifs et les géodésiques, que nous cherchons à combiner avec les méthodes d'apprentissage afin de rapprocher ces deux approches de traitement des données, dans le but de combler les lacunes d'interprétation des réseaux de neurones.

Structure de la thèse et présentation des contributions

La thèse est composée de 5 chapitres thématiques. Nous commençons par l'introduction de la thèse et de la mise en contexte. Ensuite, le deuxième chapitre présente les notions de contours actifs et de méthodes géodésiques. Nous introduisons ces notions d'un point de vue mathématique et aussi informatique pour mettre en avant leur implémentation numérique, qui nous sera utile dans la suite du manuscrit. Le troisième chapitre introduit la notion d'apprentissage profond et surtout les méthodes d'apprentissage par vision par ordinateur, ce qui correspond à l'étude en particulier des images. Le chapitre 4 présente les premiers résultats de combinaison d'apprentissage profond et de méthodes de contours actifs pour la segmentation de tumeur cérébrale. Le chapitre 5 introduit la dernière contribution sur l'étude de l'association de méthodes géodésiques avec l'apprentissage profond.

La **première contribution** que nous avons publiée permet la segmentation de tumeur cérébrales. La méthodes est présentée dans le **chapitre 4**. Le principe est de proposer une méthode permettant de mettre en œuvre deux méthodes distinctes de segmentation d'image dans un même processus pour engendrer une prediction plus sure. Les deux méthodes que nous avons employées sont l'apprentissage profond et les méthodes de contours actifs. Nous avons basé notre travail sur la notion de mécanisme d'attention dans les réseaux de neurones qui leur permettent

d'apprendre à juger de la pertinence d'éléments dans une image pour les aider à formuler une prédiction. Nous avons décidé d'aller plus loin et de contraindre ces mécanismes d'attention de résoudre un problème d'optimisation. Le problème d'optimisation à résoudre est basé sur le modèle de Chan-Vese qui permet de trouver les contours d'importances entre deux zones dans une images. L'introduction de cette méthode au cœur du processus d'apprentissage du réseau de neurones permet d'ajouter du contrôle sur l'évolution des valeurs des paramètres du réseau de neurones qui sont mis à jour également par les valeurs des dérivées provenant du processus d'optimisation. Nous présentons des résultats détaillés sur une base de données de tumeur cérébrale comprenant peu de données annotées.

La **deuxième contribution** est présentée dans le **chapitre 5**. Nous avons montré que nous pouvons intégrer les notions de distances géodésiques dans un flux d'apprentissage automatiques. L'objectif est de pouvoir apprendre automatiquement les géométries présentes dans une image sans qu'un utilisateur doive en choisir une parmi d'autres. Pour ce faire nous avons combiné la méthode de calcul de distance géodésique communément appelée Fast Marching ([Set96]), qui permet de résoudre l'équation eikonale dans un milieu, avec un réseau de neurones convolutionnel classique. L'architecture complète permet d'apprendre la métrique isotrope associée à la segmentation des tumeurs cérébrales. La segmentation est approchée en utilisant l'indicatrice de la boule unité pour la distance définie par la métrique. La possibilité de réaliser l'apprentissage est obtenue grâce à l'utilisation de la méthode de sous-gradient pour l'algorithme de Fast Marching. Cela permet d'obtenir la différenciation de la distance géodésiques par rapport aux paramètres du réseau. Les résultats obtenus sont similaires à ceux de l'état de l'art avec en plus une garantie sur le masque de segmentation.

La troisième contribution est une extension de la méthode présentée précédemment. Une description de cette contribution est présentée dans le chapitre 5 à la section 5.4.

Mise en Contexte

Le sujet général de cette thèse est l'étude et l'application de techniques informatiques avancées pour la segmentation d'images médicales. Plus précisément, nous visons à développer des méthodologies qui délimitent avec précision les frontières et identifient le contenu des images médicales, en se concentrant sur des objets tels que les tumeurs, les lésions et les réseaux vasculaires. La segmentation d'images médicales est une tâche cruciale dans le processus de diagnostic et de planification des traitements, car elle permet aux cliniciens d'obtenir des mesures et des visualisations précises nécessaires pour prendre des décisions éclairées. Au fil des ans, ce sujet a suscité une attention considérable dans les milieux universitaires et cliniques, ce qui a conduit au développement d'un large éventail d'approches. Ces approches vont des techniques mathématiques classiques, telles que la minimisation d'énergie de contours, aux avancées récentes impliquant des algorithmes assistés par ordinateur et l'apprentissage automatique. Ces dernières années, l'avènement de l'apprentissage profond a révolutionné le domaine de la segmentation des images médicales. Les techniques basées sur les réseaux neuronaux convolutifs (CNN) ont eu des succès remarquables dans l'apprentissage automatique et l'extraction de caractéristiques pertinentes à partir d'images, surpassant souvent les méthodes traditionnelles. En entraînant ces réseaux neuronaux sur de grands ensembles de données d'images médicales

annotées par des experts, les chercheurs ont atteint des performances de pointe dans diverses tâches de segmentation. Ces avancées promettent d'améliorer la précision et l'efficacité des diagnostics médicaux. Cependant, malgré les progrès significatifs réalisés dans le développement d'algorithmes de segmentation, leur adoption dans la pratique clinique doit encore être améliorée. Plusieurs défis contribuent à ce fossé entre la recherche et l'application pratique. L'une des difficultés réside dans les ressources informatiques nécessaires pour former et déployer des modèles d'apprentissage profond, en particulier dans des environnements en temps réel ou à ressources limitées. La phase d'inférence d'un réseau neuronal entraîné exige un matériel performant, qui peut n'être disponible que dans certains environnements cliniques. En outre, l'interprétabilité et la transparence des modèles d'apprentissage profond posent un autre problème. Les implications éthiques du recours à des systèmes automatisés pour des décisions médicales critiques doivent également être prises en compte : sommes-nous prêts à confier des décisions, telles que le diagnostic ou le plan de traitement d'un patient, à un modèle d'apprentissage automatique, en particulier lorsque l'enjeu concerne des vies humaines ? Bien que ces modèles puissent atteindre une précision impressionnante, leur processus de prise de décision doit souvent être plus transparent, sinon cela pose des problèmes pour la confiance des professionnels de santé. Il est essentiel de veiller à ce que ces modèles puissent fournir des résultats précis, mais aussi explicables et justifiables, pour qu'ils soient plus largement acceptés dans les milieux cliniques. La recherche en cours dans ce domaine doit relever ces défis pour combler le fossé entre les techniques de segmentation de pointe et leur mise en œuvre pratique dans les soins de santé, afin d'améliorer les résultats pour les patients et de susciter la confiance dans les systèmes automatisés.

Un Mécanisme d'Attention basée sur la Méthode de Contour Actif

Notre première contribution a été introduite dans notre article intitulé « **Chan-Vese Attention U-Net : Un mécanisme d'attention pour une segmentation robuste** » ([Mak23]), Laurent D. Cohen et moi-même y avons présenté une nouvelle approche pour améliorer la segmentation des images médicales en combinant l'apprentissage profond avec des techniques classiques de minimisation de l'énergie. La segmentation des images médicales est une tâche cruciale qui exige souvent un effort important de la part des professionnels de la santé. Bien que les réseaux neuronaux convolutifs (CNN), en particulier les architectures comme U-Net ([Ron15]), se soient montrés très prometteurs dans l'automatisation de ce processus, leur application en milieu clinique soulève encore des inquiétudes quant à leur fiabilité, leur transparence et leur facilité d'utilisation. Dans cet article, nous avons proposé une nouvelle méthode qui intègre un mécanisme d'attention basé sur la minimisation de l'énergie de Chan-Vese ([Cha01]) dans l'architecture U-Net afin d'améliorer la précision et la fiabilité de la segmentation. Nous avons passé en revue diverses tentatives d'intégration de propriétés géométriques et topologiques dans les réseaux neuronaux pour les tâches de segmentation. Les approches précédentes, telles que les méthodes de contour actif et les contours actifs géodésiques intégrés aux CNN, ont contribué à l'amélioration de la segmentation, mais nous avons constaté un potentiel d'intégration plus efficace de ces techniques classiques avec les méthodes modernes d'apprentissage en profondeur.

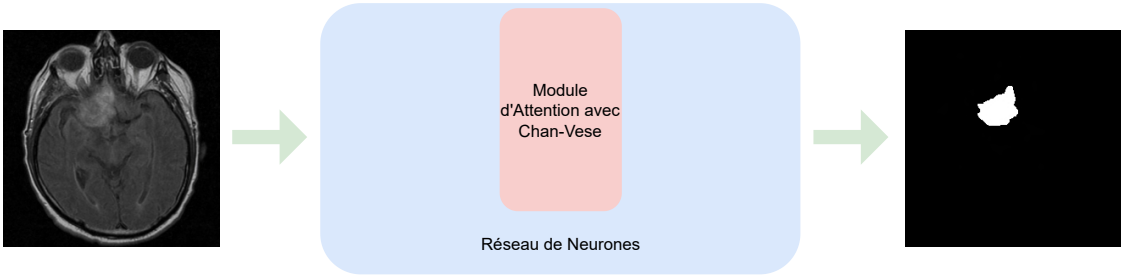


Figure 1: Schéma illustrant le processus de segmentation de tumeur cérébrale à partir d'une image IRM. L'image est d'abord traitée par un réseau de neurones, complété par un module d'attention basé sur le modèle de Chan-Vese, pour obtenir une segmentation précise de la tumeur (en blanc).

Notre principale contribution est le développement de la porte d'attention de Chan-Vese (voir Figure 1), un nouveau mécanisme d'attention qui exploite les informations de la fonctionnelle de Chan-Vese, une méthode de minimisation de l'énergie bien établie. Ce mécanisme est conçu pour affiner la segmentation en incorporant efficacement des informations spatiales, offrant un contrôle plus précis sur le processus de segmentation que les approches traditionnelles d'apprentissage profond. Nous pensons que cette méthode pourrait conduire à des résultats plus précis et plus cohérents en imagerie médicale. Le mécanisme d'attention consiste à appliquer l'algorithme de Chan-Vese, qui est traditionnellement utilisé pour les modèles de contours actifs sans bord. Dans notre implémentation, l'algorithme de Chan-Vese est adapté pour fonctionner dans un environnement de réseau neuronal pour affiner le masque de segmentation généré à partir de la transformée de distance. La méthode de Chan-Vese combine des aspects des contours actifs et de la fonctionnelle de Mumford-Shah ([Mum89]) pour faire évoluer un contour qui minimise une fonctionnelle d'énergie, segmentant efficacement l'image en régions d'intérêt. En incorporant cette méthode dans notre réseau, nous tirons parti de sa capacité à imposer des contraintes à la fois globales et locales sur la segmentation, en veillant à ce que les masques résultants soient non seulement précis, mais qu'ils respectent également les propriétés géométriques attendues. Cette étape permet de s'assurer que le réseau ne repose pas uniquement sur les gradients d'intensité ou la contiguïté des pixels, mais qu'il intègre également des informations géométriques d'ordre supérieur qui peuvent conduire à des résultats de segmentation plus fiables. L'algorithme modifié utilise le contour initial fourni par la transformée de distance comme point de départ. La fonction d'énergie de Chan-Vese est ensuite minimisée d'une manière qui respecte la nature différentiable du cadre d'apprentissage profond. Cela implique de modifier l'algorithme classique de Chan-Vese pour s'assurer que le processus de minimisation de l'énergie peut être rétro propagé à travers le réseau. Cette modification est essentielle car elle permet au réseau d'apprendre du processus de segmentation lui-même, en améliorant continuellement sa capacité à produire des masques de segmentation précis et fiables Nous avons mené des expériences en utilisant la base de données TCGA-LGG ([Ped]), qui contient des IRM de patients atteints de tumeurs cérébrales. Nous avons comparé notre

Chan-Vese Attention U-Net au U-Net traditionnel et à un U-Net d'attention. Nos résultats ont montré que le Chan-Vese Attention U-Net a obtenu de meilleurs scores d'Intersection Over Union (IOU) et de meilleurs taux de faux négatifs, ce qui indique une meilleure précision et fiabilité de la segmentation. Nous avons également analysé les masques d'attention générés par le Chan-Vese Attention Gate. Les résultats ont démontré que le masque d'attention converge rapidement vers une segmentation ressemblant étroitement à la région tumorale, en affinant les contours au fur et à mesure de l'entraînement. Ce comportement contraste avec les mécanismes d'attention traditionnels, qui peuvent inclure des artefacts non pertinents en dehors de la zone tumorale. La focalisation de notre méthode sur la région d'intérêt à l'intérieur du crâne a conduit à une segmentation plus précise et plus fiable (voir Figure 1.

Apprentissage de la Métrique Riemannienne

Notre deuxième contribution a été introduite dans notre article intitulé « **Fast Marching Energy CNN** » ([Ber23]), mes co-auteurs et moi-même y avons présenté une nouvelle approche de la segmentation d'images en intégrant le calcul des distances géodésiques aux réseaux neuronaux. L'idée est d'exploiter l'information géométrique véhiculée par les distances géodésiques pour améliorer la segmentation des images médicales, en particulier des tumeurs cérébrales. Les distances et les courbes géodésiques sont utilisées depuis longtemps pour représenter les propriétés géométriques dans diverses applications d'imagerie([Sap95; Pey10; Che14]). Traditionnellement, ces méthodes s'appuient sur des connaissances préalables pour définir explicitement une métrique Riemannienne à partir de l'image. Cependant, notre approche élimine la nécessité d'une telle définition manuelle de la métrique. Nous proposons plutôt de générer une métrique Riemannienne isotrope directement à partir des données à l'aide d'un réseau neuronal, entraîné de manière supervisée. Cette approche réduit les biais de l'utilisateur et la nécessité de régler les paramètres, ce qui rend le processus de segmentation plus simple et plus efficace. La distance géodésique a une riche histoire dans les tâches de segmentation d'images. Les premières méthodes, comme celles de Malladi et al. [Mal98], utilisaient les distances géodésiques pour segmenter les images cérébrales en 3D, la distance géodésique aide le contour à trouver le chemin le plus court vers les bords de l'objet à segmenter tout en tenant compte de la structure de l'image. Des études ultérieures ([Che18; Che16; Yan16]) ont développé ces idées en introduisant des métriques anisotropes et en s'adaptant à des tâches spécifiques, telles que la segmentation des structures vasculaires. Bien que ces méthodes se soient avérées efficaces, elles ne traitent généralement pas la tâche de segmentation de manière holistique ou ne se généralisent pas bien à de grands ensembles de données.

Seules quelques méthodes récentes ont exploré l'apprentissage d'une métrique à partir de données, comme les travaux de Scarvelis et al. [Sca22] et Heitz et al. [Hei21], qui visent à trouver des tenseurs métriques s'adaptant aux données spatio-temporelles pour capturer les champs de vitesse et la géométrie. Cependant, ces approches ne généralisent pas complètement la génération de tenseurs métriques, et c'est là que notre méthode comble l'écart. La principale contribution de notre travail est l'intégration d'un réseau neuronal avec le calcul de la distance géodésique pour la segmentation d'images. Notre approche utilise une architecture U-Net modifiée pour générer des masques de segmentation sous forme de boules géodésiques,

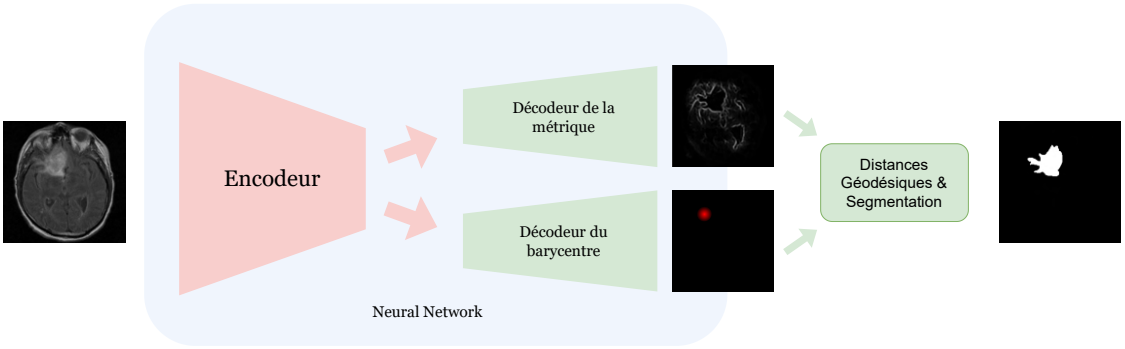


Figure 2: L'encodeur traite les caractéristiques de l'image, tandis que les décodeurs respectifs extraient la métrique géodésique et le barycentre pour guider la segmentation.

déterminées par une métrique apprise et un centre ou point d'amorçage. Ce cadre nous permet d'appliquer des contraintes géométriques et topologiques sur le masque de sortie, ce qui se traduit par des segmentations plus précises et plus fiables. La distance géodésique mesure la longueur du chemin le plus court entre deux points d'une variété qui, dans ce cas, est le domaine de l'image. Nous calculons cette distance à l'aide de l'algorithme Fast Marching [Set96], une méthode bien connue de propagation des fronts qui résout l'équation Eikonale. La distance géodésique est l'unique solution de viscosité positive de l'équation.

Pour incorporer la distance géodésique dans notre cadre de réseau neuronal, nous avons dû la différencier par rapport à la métrique, une technique introduite pour la première fois par Benmansour et al. [Ben10]. Nous appliquons cette différenciation dans notre cadre d'apprentissage profond à l'aide d'une méthode connue sous le nom d'algorithme de marche de sous-gradient.

Le modèle que nous proposons utilise une architecture U-Net [Ron15] modifiée avec deux voies de décodage distinctes. Le premier décodeur prédit la métrique requise pour le module de Fast Marching, tandis que le second décodeur estime un potentiel Gaussien représentant la probabilité du barycentre de la région. Le module Fast Marching calcule ensuite une carte de distance basée sur ces prédictions, ce qui nous permet de générer des boules géodésiques pour la segmentation. La segmentation finale est obtenue en comparant le masque prédit à la vérité terrain et en ajustant en fonction des erreurs de segmentation et de prédiction du barycentre (Voir Figure 2.

Le module Fast Marching intégré dans notre cadre a conduit à une convergence plus rapide et à une précision de segmentation améliorée, en particulier dans la détection des bords. Notre

méthode a toujours produit des contours bien définis en tenant compte de la morphologie de l'image, une capacité que les filtres traditionnels n'ont pas.

En résumé, les contributions présentées dans cette thèse visent à améliorer la segmentation des images médicales en combinant des techniques classiques de géométrie et de traitement d'images avec des approches modernes d'apprentissage profond. Ces travaux ouvrent de nouvelles perspectives pour développer des outils plus précis et fiables, essentiels pour l'analyse et le diagnostic médicaux.

Chapter 1

Introduction

The intersection of humanism and artificial intelligence (AI) technology raises important questions about the place of the human in a world increasingly inhabited by non-human entities. In this respect, typography offers an illuminating metaphor. Traditionally conceived as the 'servant of meaning', typography transcends its apparent dumbness by playing with full and empty, black and white. This silent game gives rise to signs and symbols beyond the printed text.

However, when assessing the text-generating capabilities of machines like GPT-4, one question stands out: can these algorithms reproduce the polysemy and eloquent silence that are the hallmarks of human communication? At first glance, AI models seem formidable in signal processing, whether textual or graphical. But it is essential to note that these models are trained on explicit content and do not perceive what is implicit, what surrounds the signal: the white space in typography, the silences in music or the unspoken words in a conversation.

The mismatch between AI and these more subtle forms of communication highlights the intrinsic limitations of current models. At a time when AI is increasingly capable of simulating human behaviour, it is crucial to question what fundamentally distinguishes man from machine. It is also a reminder that humanism, the ideology that places humans at the centre, faces an existential challenge. Are we still unique in the world of symbols and signs?

Humanism is being forced to rethink its centrality in its dialogue with AI. We may be at the dawn of a new age in which cohabitation with competent non-human entities changes our understanding of what is 'human'. However, it remains to be seen whether these entities will ever be able to grasp or reproduce the nuances of polysemy or 'eloquent silence' that enrich our subjective experience. It may be precisely in these nuances that our human specificity resides.

Overview of Medical Image Segmentation

The general topic of this thesis is the study and application of advanced computational techniques for medical image segmentation. Specifically, we aim to develop methodologies that accurately delineate boundaries and identify content within medical images, focusing on objects such as tumours, lesions, and vascular networks. Medical image segmentation is a crucial

task in the diagnostic and treatment planning process, as it allows clinicians to obtain precise measurements and visualisations necessary for making informed decisions. Over the years, this topic has attracted considerable attention in academic and clinical settings, leading to the development of a wide range of approaches. These approaches span from classical mathematical techniques, such as energy minimisation frameworks, to recent advancements involving computer-assisted algorithms and Machine Learning.

In recent years, the advent of Deep Learning has revolutionised the field of medical image segmentation. Techniques based on convolutional neural networks (CNNs) have shown remarkable success in automatically learning and extracting relevant features from images, often outperforming traditional methods. By training these neural networks on large datasets of annotated medical images, where experts perform the segmentation, researchers have achieved state-of-the-art performance in various segmentation tasks. These advancements promise to enhance the accuracy and efficiency of medical diagnoses.

However, despite the significant progress made in developing segmentation algorithms, their adoption in clinical practice still needs to be improved. Several challenges contribute to this gap between research and practical application. One major challenge is the computational resources required to train and deploy Deep Learning models, particularly in real-time or resource-constrained environments. The inference phase of a trained neural network demands high-performance hardware, which may only be readily available in some clinical settings.

Furthermore, the interpretability and transparency of Deep Learning models pose another concern. The ethical implications of relying on automated systems for critical medical decisions must also be considered whether we are prepared to entrust decisions, such as a patient's diagnosis or treatment plan, to a machine-learning model, especially when the stakes involve human lives. While these models can achieve impressive accuracy, their decision-making process often needs to be more transparent, leading to challenges in gaining the trust of medical professionals. Ensuring that these models can provide accurate but also explainable and justifiable outputs is crucial for their broader acceptance in clinical settings. The ongoing research in this field must address these challenges to bridge the gap between state-of-the-art segmentation techniques and their practical implementation in healthcare, ultimately improving patient outcomes and creating trust in automated medical systems.

Research Landscape: A Literature Review

The issue of transparency in AI decision-making, particularly in the context of medical applications, has caught significant attention, leading to the development of multiple methods aimed at making AI systems more interpretable and trustworthy. Addressing this challenge is crucial, as the nature of Deep Learning algorithms can hinder their adoption in domains such as healthcare, where decisions have important implications for patient outcomes.

One of the approaches to improving transparency involves decomposing the prediction process into two distinct stages. The first stage focuses on predicting human-level concepts that drive decision-making among clinical experts, and the second stage uses these concepts to produce the final decision. This methodology mirrors the decision-making process of medical professionals, thereby enhancing the interpretability of the model's outputs. For instance, in the

work of Koh et al. [Koh20], a Deep Learning algorithm was developed to predict the presence of arthritis by first identifying specific clinical concepts such as bone spurs, calcification, and joint space narrowing—factors commonly used by clinicians to assess the severity of arthritis. These intermediate predictions are then aggregated to determine the stage of arthritis, making the model's decision-making process more transparent and aligned with clinical reasoning.

Further advancements in this area have introduced methods that encode high-level visual attributes within vector representations, as demonstrated by LaLonde et al. [LaL20]. In this approach, the model encodes specific radiological features, such as the shape, margin, and texture of lung nodules, into vectors corresponding to radiologists' attributes in diagnosing lung cancer. By making these features explicit, the model's predictions become more interpretable, allowing clinicians to understand the basis of the AI's decision regarding familiar diagnostic criteria.

Another promising approach is using prototypes to provide global and local explanations, as proposed by Kim et al. [Kim21] in their XProtoNet framework. Prototypes are representative patterns of diseases learned from a dataset of X-ray images. The model diagnoses a given image by comparing its features with the learned prototypes. This comparison provides a visual and conceptual explanation of the model's decision, showing how similar the current case is to previously learned examples. The advantage of this approach lies in its flexibility, allowing the model to learn and use relevant patterns dynamically. It can adapt its explanations to the specific characteristics of each disease case.

In addition to these concept-driven approaches, visual methods have been widely adopted to make the decision-making process of Deep Learning models more straightforward. Among these, Grad-CAM from Selvaraju et al. [Sel17] is one of the most commonly used techniques. Grad-CAM generates visual explanations by computing the gradient of the target class with respect to the feature maps of a convolutional layer. It highlights the areas of the input image that most influence the model's prediction. This technique has been effectively used in various medical imaging studies to reveal which regions of an image the model focuses on when making a prediction. For example, Pereira et al. [Per18] employed Grad-CAM to analyse the predictions of CNN models applied to brain MRI images, specifically in the context of tumour grading. The Grad-CAM heatmaps generated in their study provided insights into whether the model has correctly identified relevant tumour regions or is being misled by irrelevant artefacts, thereby offering a means of assessing and improving model reliability.

Beyond pixel-level explanations, there is a growing interest in concept-level explanations, which aim to abstract the decision-making process further by associating predictions with high-level, user-defined concepts. The framework proposed by Graziani et al. [Gra20] lets users define concepts relevant to a particular medical diagnosis, which the Deep Learning model predicts as scores. These concept scores are subsequently used to calculate, for instance, the probability of a malignant tumour. Importantly, this framework allows the model to quantify the contribution of each concept to the final decision, offering a clear and interpretable explanation behind the model's predictions. This approach also improves transparency by shifting the focus from raw pixel values to clinically meaningful concepts to propose another solution to the transparency problem. It aligns the AI's reasoning process more closely with human expertise.

Key Concepts and Terminologies

Medical image segmentation is a critical process in medical imaging, where the objective is to partition an image into distinct regions that correspond to different anatomical or pathological structures. This process is essential for various clinical applications, including diagnosis, treatment planning, and disease monitoring. Segmentation allows clinicians to isolate specific areas of interest, such as tumours, lesions, or vascular networks, from surrounding tissues, enabling precise measurements, visualisations, and analyses crucial for making informed medical decisions.

"Medical image segmentation" refers to the computational techniques ([Nor14]) employed to achieve this partitioning. These techniques range from traditional mathematical modelling and energy minimisation methods to advanced Machine Learning algorithms automatically learning to segment images based on large datasets. The segmentation task can be challenging due to the complexity and variability of medical images, which may be affected by noise, artefacts, and differences in patient anatomy.

This thesis focuses specifically on using Deep Learning techniques for medical image segmentation. We illustrate mainly using MRI (Magnetic Resonance Images) data to segment brain tumours. Deep Learning, a subset of Machine Learning, involves training neural networks with multiple layers to learn hierarchical representations of data automatically. In medical image segmentation, Deep Learning models, particularly convolutional neural networks (CNNs), have demonstrated remarkable success in identifying and delineating structures within medical images, often achieving results that surpass traditional methods.

Advancements, Challenges, and Future Directions in Medical Image Segmentation

The field of medical image segmentation has experienced significant advancements over the past few decades, particularly with the rise of Machine Learning and Deep Learning techniques ([Wan22; Alz21]). Traditionally, segmentation tasks were approached using classical methods such as thresholding, region growing, and edge detection, often combined with mathematical frameworks like active contours and level sets. While these methods provided a solid foundation, they struggled to handle the complex variability and subtle differences in medical images. It leads to challenges in achieving accurate and reliable segmentation.

The new Machine Learning methods marked a pivotal shift in the field, enabling more automated and data-driven approaches to segmentation. Early Machine Learning methods relied on handcrafted features and shallow classifiers, which, despite improvements over traditional techniques, still faced limitations in generalisation across datasets and imaging modalities. The introduction of Deep Learning, particularly convolutional neural networks (CNNs) ([Sar22]), further enhanced medical image segmentation by allowing models to automatically learn and extract hierarchical features directly from raw image data.

In recent years, Deep Learning models have become the dominant approach in medical image segmentation. CNN-based architectures, such as U-Net and its variants, have set new benchmarks in accuracy and efficiency across various segmentation tasks—from identifying

tumours in MRI and CT scans to segmenting organs and blood vessels in ultrasound images. These models handle texture, shape, and intensity variations common in clinical datasets. Their scalability allows them to be applied to large-scale datasets, allowing the development of robust, high-performing systems.

Despite these advancements, several challenges remain ([Raz18]). One of the primary challenges is the dependence on large annotated datasets. High-quality annotations, typically performed by medical experts, are essential for training Deep Learning models. However, this process is labour-intensive, time-consuming, and expensive. It creates bottlenecks in developing and deploying these models. Moreover, the small number of labelled data for specific medical conditions or imaging modalities can limit the models' generalizability and effectiveness.

Another critical disadvantage is Deep Learning models' "black box" nature. Although techniques have been developed to improve interpretability, the complexity of these models often makes it difficult to understand how they arrive at specific decisions. This lack of transparency can be a significant barrier to clinical adoption, as medical professionals may be reluctant to trust a system they cannot fully explain and understand, especially when patient outcomes are at stake.

In response to these challenges, researchers have developed various approaches to enhance the transparency and interpretability of Deep Learning models. Techniques such as concept-based models aim to mimic the decision-making process of clinicians by predicting human-level concepts that are then used to make final predictions. Prototype-based methods provide explanations by comparing input images to representative examples the model has learned, offering global and local insights into the model's decision-making process. Visual tools like Grad-CAM are also widely used to generate heatmaps highlighting the regions of an image the model considers most important for its prediction, offering a form of visual validation clinicians can evaluate.

While progress has been made in improving the interpretability of Deep Learning models, a trust gap still exists between AI systems and clinicians. More work is needed to develop models that perform well and offer intuitive and aligned explanations with clinical practice. This could involve integrating more sophisticated explanation methods beyond current visual and prototype-based techniques, possibly incorporating domain-specific knowledge or reasoning based on more classical methods with provable guarantees.

Relevance and Potential Impact

The proposed research addresses critical challenges in applying Deep Learning to medical image segmentation, a process essential for accurate diagnosis and treatment planning in healthcare. Despite the significant progress made with Deep Learning models, obstacles still limit their practical use in clinical settings.

One major challenge is the need for AI models that are not only accurate but also easy to understand. In a clinical environment, doctors and other healthcare professionals must be able to trust and comprehend the decisions made by AI systems, significantly when these decisions impact patient care. However, many current Deep Learning models operate as "black boxes,"

making explaining how they reach their conclusions difficult. This lack of transparency can hinder the adoption of these models in everyday medical practice.

This research is necessary because it aims to develop Deep Learning models that are both interpretable and less dependent on extensive annotated datasets. By exploring attention-based and mathematically-based approaches, the research seeks to create models that provide more apparent and trustable outputs, helping healthcare professionals make informed decisions.

Research Questions and Objectives

The research problem focuses on the challenges associated with the practical application of Deep Learning models for medical image segmentation in clinical settings. Specifically, the issue concerns more interpretability in current deep-learning models. Research Questions:

1. **How can Deep Learning models for medical image segmentation be made more interpretable to ensure their decision-making processes are reliable?**

    - This question explores ways to make AI models more transparent, allowing clinicians to understand and trust the outputs provided by these systems. The focus is on developing methods that explain the model's decisions in a meaningful and useful way for medical professionals.

2. **How can energy-based methods, which have demonstrated proven convergence properties, be integrated into Deep Learning models for medical image segmentation to enhance the reliability and interpretability of predicted features?**

    - This question explores the potential of combining traditional energy-based techniques with Deep Learning models to improve the reliability of segmentation outputs. The focus is on leveraging the convergence properties of energy-based methods to provide a more stable and interpretable feature extraction process within Deep Learning frameworks.

3. **How can the propagation of the geodesic front, as modelled by fast marching methods, be incorporated into Deep Learning models to enforce a more structured and interpretable decision-making process in medical image segmentation?**

    - This question investigates the use of geodesic front propagation techniques within Deep Learning models to introduce a geometric understanding of the image, guiding the model's decision-making process. The goal is to see if this approach can lead to more interpretable outcomes by enforcing decisions that align with the inherent structure of the medical images.

Structure and Organization of the Study

The thesis is divided into five thematic chapters. We begin by introducing the thesis and setting the context. Then, the second chapter presents the notions of active contours and geodesic

methods. We introduce these notions from a mathematical and computational point of view to highlight their numerical implementation, which will be helpful to us in the remainder of the manuscript. The third chapter introduces the notion of Deep Learning, especially computer vision learning methods, which correspond to the study of images. Chapter 4 presents the first results combining Deep Learning and active contour methods for brain tumour segmentation. Chapter 5 introduces the last contribution to the study of the association of geodesic methods with Deep Learning.

Our **first contribution** was introduced in our paper entitled **'Chan-Vese Attention U-Net: An attention mechanism for robust segmentation'** ([Mak23]), in which Laurent D. Cohen and I presented a new approach to improving the segmentation of medical images by combining deep learning with classic energy minimisation techniques. The method is presented in the **chapter 4**.

The segmentation of medical images is a crucial task that often requires much effort from healthcare professionals. Although convolutional neural networks (CNNs), in particular architectures such as U-Net ([Ron15]), have shown great promise in automating this process, their application in clinical settings still raises concerns about their reliability, transparency and ease of use. This paper proposes a new method that integrates an attention mechanism based on Chan-Vese energy minimisation ([Cha01]) into the U-Net architecture to improve segmentation accuracy and reliability. We have reviewed various attempts to integrate geometric and topological properties into neural networks for segmentation tasks. Previous approaches, such as active contour methods and geodesic active contours embedded in CNNs, have improved segmentation. Still, we have seen the potential for more effective integration of these classic techniques with modern deep learning methods.

Our main contribution is the development of the Chan-Vese attention gate (see Figure 1.1), a novel attention mechanism that exploits information from the Chan-Vese functional, a well-established energy minimisation method. This mechanism is designed to refine segmentation by efficiently incorporating spatial information, offering more precise control over the segmentation process than traditional deep learning approaches. We believe this method could lead to more accurate and consistent results in medical imaging. The attention mechanism applies the Chan-Vese algorithm, which is traditionally used for edge-free active contour models. In our implementation, the Chan-Vese algorithm is adapted to work in a neural network environment to refine the segmentation mask generated from the distance transform. The Chan-Vese method combines aspects of active contours and the Mumford-Shah functional ([Mum89]) to evolve a contour that minimises an energy functional, efficiently segmenting the image into regions of interest. By incorporating this method into our network, we take advantage of its ability to impose global and local constraints on the segmentation, ensuring that the resulting masks are accurate and respect the expected geometric properties. This step ensures that the network does not rely solely on intensity gradients or pixel adjacency but also incorporates higher-order geometric information that can lead to more reliable segmentation results. The modified algorithm uses the initial contour provided by the distance transform as a starting point. The Chan-Vese energy function is then minimised in a way that respects the differentiable nature of the deep learning framework. This involves modifying the classic Chan-Vese algorithm to ensure the energy minimisation process can be back-propagated through the network. This

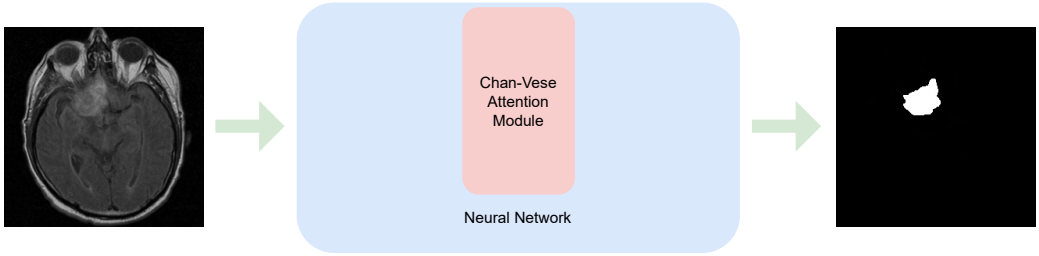


Figure 1.1: Schematic diagram illustrating segmenting a brain tumour from an MRI image. The image is first processed by a neural network, supplemented by an attention module based on the Chan-Vese model, to segment the tumour (in white) accurately

.

modification is essential as it allows the network to learn from the segmentation process, continually improving its ability to produce accurate and reliable segmentation masks. We conducted experiments using the TCGA-LGG ([Ped]) database, which contains MRI scans of brain tumour patients. We compared our Chan-Vese Attention U-Net to the traditional U-Net and an attention U-Net. Our results showed that the Chan-Vese Attention U-Net achieved better Intersection Over Union (IOU) scores and false negative rates, indicating better segmentation accuracy and reliability. We also analysed the attention masks generated by the Chan-Vese Attention Gate. The results showed that the attention mask rapidly converges on a segmentation closely resembling the tumour region, refining the contours as training progresses. This behaviour contrasts with traditional attention mechanisms, which can include irrelevant artefacts outside the tumour area. Focusing our method on the region of interest within the skull led to more accurate and reliable segmentation (see Figure 1.1).

Our **second contribution** was introduced in our article entitled **'Fast Marching Energy CNN'** ([Ber23]), in which my co-authors and I presented a new approach to image segmentation by integrating the calculation of geodesic distances with neural networks. The idea is to exploit the geometric information conveyed by geodesic distances to improve the segmentation of medical images, particularly brain tumours. The method is presented in the **chapter 5**.

Geodesic distances and curves have long been used to represent geometric properties in various imaging applications([Sap95; Pey10; Che14]). Traditionally, these methods rely on prior knowledge to explicitly define a Riemannian metric from the image. However, our approach eliminates the need for such a manual metric definition. Instead, we propose to generate an isotropic Riemannian metric directly from the data using a neural network trained in a supervised manner. This approach reduces user bias and the need for parameter tuning, making the segmentation process simpler and more efficient. Geodesic distance has a rich history in image segmentation tasks. Early methods, such as those by Malladi et al. [Mal98], used geodesic distances to segment 3D brain images. The geodesic distance helps the contour find the shortest path to the edges of the object to be segmented while taking into account the structure of the image. Later studies ([Che18; Che16; Yan16]) developed these ideas by introducing anisotropic metrics and adapting them to specific tasks, such as segmenting vascular structures. Although these methods have proved effective, they generally do not treat the segmentation task holistically or generalise well to large datasets.

Only a few recent methods have explored learning a metric from data, such as the work of Scarvelis et al. [Sca22] and Heitz et al. [Hei21], which aim to find metric tensors that adapt to spatio-temporal data to capture velocity fields and geometry. However, these approaches still need to generalise the generation of metric tensors fully, and this is where our method fills the gap. The main contribution of our work is integrating a neural network with geodesic distance computation for image segmentation. Our approach uses a modified U-Net architecture to generate segmentation masks in the form of geodesic balls determined by a learned metric and a centre or seed point. This framework allows us to apply geometric and topological constraints on the output mask, resulting in more accurate and reliable segmentations. The geodesic distance measures the shortest path length between two points of a variety, which in this case is the image domain. We calculate this distance using the Fast Marching [Set96] algorithm, a

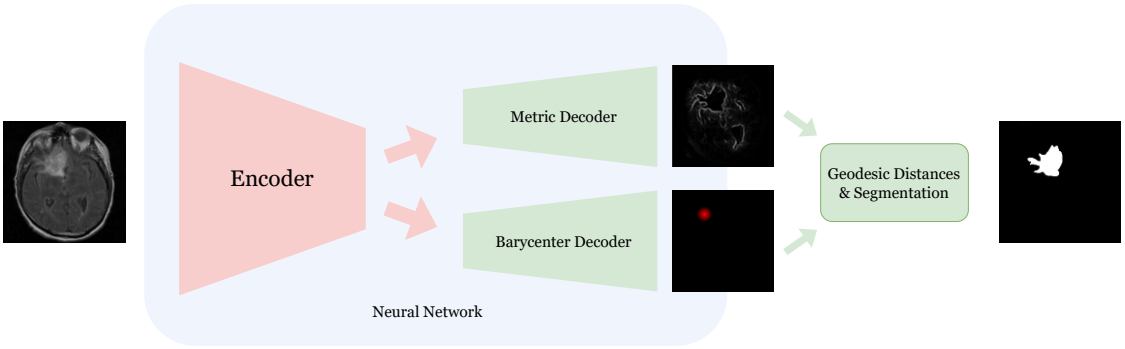


Figure 1.2: The encoder processes the image features, while the respective decoders extract the geodesic metric and barycenter to guide the segmentation.

well-known front propagation method that solves the Eikonal equation. The geodesic distance is the only positive viscosity solution to the equation.

To incorporate the geodesic distance into our neural network framework, we had to differentiate it from the metric, a technique first introduced by Benmansour et al. [Ben10]. We apply this differentiation in our deep learning framework using a method known as the subgradient walk algorithm.

Our proposed model uses a modified U-Net architecture [Ron15] with two distinct decoding paths. The first decoder predicts the metric required for the Fast Marching module, while the second decoder estimates a Gaussian potential representing the probability of the region's barycentre. The Fast Marching module then calculates a distance map based on these predictions, allowing us to generate geodesic balls for segmentation. The final segmentation is obtained by comparing the predicted mask with the ground truth and adjusting for segmentation and barycentre prediction errors (see Figure 1.2).

The Fast Marching module integrated into our framework led to faster convergence and improved segmentation accuracy, particularly in edge detection. Our method consistently produced well-defined edges by taking into account the morphology of the image, a capability that traditional filters lack.

The third contribution is an extension of the method presented above. A description of this contribution is presented in the chapter 5 in section 5.4.

Chapter 2

Technical Background on Active Contours and Geodesic Methods

Objectifs

This chapter provides a comprehensive examination of active contour models, which have significantly influenced the field of image segmentation. We discuss their widespread adoption in various fields, such as medical image segmentation, object tracking, and 2D or 3D image reconstruction. Our analysis reveals the fundamental elements of active contour models and their strengths and weaknesses. We then shift our focus to level-set methods in computational geometry. The aim is not to identify application domains for specific contours but to offer an insightful understanding of the diverse active contour models and level-set methods.

Contents

Since their appearance in [Kas88], active contour models have had several successes, making them very popular. Many fields have turned to them for different tasks such as medical image segmentation [Coh96; Coh01], object tracking [Bar96], 2D or 3D curve or surface reconstruction [Coh90; Coh93a].

The wide adoption of active contour methodologies for segmentation purposes has fostered many models available. This rich assortment facilitates the opportunity to approach segmentation from varied perspectives. Each model focuses on a different challenge. For example, while some models might focus on speed in computational execution, others might emphasise precision in segmentation tasks. Specific models may perform efficiently when dealing with images consisting of relatively homogeneous areas but may need to improve when faced with noisy or texture-rich images.

In this chapter, we explore various active contour models in depth. Our principal aim is to provide a thorough overview of the field of active contour models, concentrating on a detailed discussion of fundamental models that have significantly impacted this specific facet of image segmentation.

Our examination of the active contour is divided into two distinct sections. In the first section, we will unveil the fundamental blueprint of an active contour without restricting our discourse to particular models. Our analysis will reveal that an active contour model fundamentally consists of two primary elements: an energy function and a mode of representation (how to solve the problem). Following this, we will probe the intrinsic properties of each contour type, seeking to understand their strengths and weaknesses and to spotlight key active contour models that have substantially influenced the field of image segmentation. We do not intend to identify application domains where a specific contour may demonstrate effectiveness; instead, we want to present a comprehensive understanding of different contour model categories.

In the second section, we focus on the domain of level-set methods. With its roots firmly planted in the broader field of computational geometry, this technique has been an instrumental tool in image processing and computer vision, offering a flexible approach to handling deformable shapes. We will examine the fundamental principles underpinning level-set methods, tracing their evolution and impact on the landscape of image segmentation.

Afterwards, we focus on geodesic methods. This section introduces important concepts like the geodesic distance, the Eikonal equation, the Fast Marching method, and the distance transform.

2.1 Active Contours Model

The concept of active contours, initially introduced by Kass et al. [Kas88], has been the subject of intense study and exploration over the past 35 years. This interest has led to variations and adaptations of these deformable models. The robustness and versatility of active contours have made them a popular choice for a myriad of applications, spanning from image segmentation to object tracking [Fuj93; Ley93; Del95] and beyond.

Snakes, or active contour models, are used in computer vision research to accurately localise nearby edges and features. They are different from other edge detection and feature

localisation methods because they are guided by external constraint forces and influenced by image forces. Snakes can be used for various visual problems, such as edge detection, motion tracking, and stereo matching. They provide a unified account of these problems, allowing for a more efficient and practical approach to visual analysis.

Active contour models are designed to delineate the boundaries of a specific region within an image. The approach involves defining a curve that minimises a particular function. In practice, this function is essentially a sum of energy terms that strike a balance between the smoothness of the shape and the contour's delineation by the image's gradient. In other words, the goal is to find a curve that is, on the one hand, as smooth and as regular as possible and, on the other hand, aligned with the high gradient regions of the image that usually correspond to the boundaries. This dual objective is solved by formulating the problem as a function of an internal energy, a smoothness term, and an external energy, an image term.

2.1.1 Model Definition

In this section, we describe the mathematical formulation of the active contour model in a 2D context, which applies to images.

We define an image I on a domain $\Omega \in \mathbb{R}^2$. This means that our image I is a function that assigns an intensity value to each point in the two-dimensional domain Ω. Next, we introduce a regularised curve, also known as a snake, denoted as $\boldsymbol{\gamma}$. :

$$\boldsymbol{\gamma} : [0,1] \to \mathbb{R}^2 \tag{2.1}$$

$$s \mapsto \boldsymbol{\gamma}(s) \tag{2.2}$$

$$\text{with } \boldsymbol{\gamma}(0) = a \in \Omega \text{ and } \boldsymbol{\gamma}(1) = b \in \Omega. \tag{2.3}$$

We can have two types of contours: one that is open if $a \neq b$ and one that is closed if $a = b$. For the rest of the chapter, we will focus on the case where we have a closed contours s.t. for $x \in [0,1]$ we have $\boldsymbol{\gamma}(0) = \boldsymbol{\gamma}(1)$ to ensure that the curve is closed and $\boldsymbol{\gamma} \in \mathcal{C}^2$ to have a smooth curve and well-defined curvature (See Figure 2.1).

The contour $\boldsymbol{\gamma}$ is initialised on the image domain Ω, and it moves in the direction of the normal and tangent vector. At each time step t, the contour evolves to minimise an energy functional, which, together with the image data, will govern the geometric behaviour of the model. As the contour is to evolve during the minimisation, we represent the set of all contours as $\boldsymbol{\gamma}(s,t)$.

To address the problem, one can formulate it as the minimisation of an energy to use gradient descent, for example.

We aim at finding $\boldsymbol{\gamma}$ such that the following functional is minimised :

$$E(\boldsymbol{\gamma}) = \int_0^1 E_{int}(\boldsymbol{\gamma})(s)) + E_{ext}(s) ds. \tag{2.4}$$

This functional represents the total energy of the curve, and it is a sum of two terms:

1. The internal energy E_{int}, which accounts for the smoothness of the curve, governs the

geometric evolution of the curve, and usually it is defined as a perimeter, an area or a curvature;

2. The external energy E_{ext} is the data fidelity term that measures how well the curve aligns with the image features. This energy term gives access to two categories of active contour methods: contour-based and region-based.

The energy functional E is a marker indicating how accurately the contour has performed segmentation and isolated the object of interest from an image. Therefore, when the contour has aligned with the boundaries of the object we are interested in, we should have E at its minimum. To reach this state, the contour will gradually undergo a series of energy minimisations until it stabilises to a point where the curve's energy is at a local minimum. The energy functional tied to the contour substantially shapes the model's potential for segmentation. It should then reflect the desired behaviours. We can broadly categorise functions into two main types: contour-based functions and region-based functions.

Only the information along the contour C is considered during the evolution process when dealing with contour-based active contours. Generally, these criteria are grounded on the intensity gradient of the pixels in the image (for instance, using the inverse of the gradient norm, the functional can approach zero for significant disparities in image intensity). However, this measurement is integrated solely along the curve. While effective, contour-based active contours have a substantial vulnerability to noise, limiting their use to images where simple gradients define boundaries between distinct objects. Consequently, their application becomes more nuanced in intricate, textured images.

Conversely, region-based active contours use a broader spectrum of image information, employing criteria referred to as region descriptors. These descriptors are primarily characterised by the traits of the region enclosed by C. They can manifest in various forms, such as mean or standard deviation of pixel intensities, intensity histograms, texture descriptors, etc. Upon inclusion in the energy functional, a region-based data attachment criterion is not just integrated along the curve but also across the area enclosed by it [Zhu96; Par02; Coh93a] and sometimes even over the entire image [Cha01; Ves02]. Consequently, while region-based active contours boast superior segmentation capabilities, their effectiveness depends on selecting the region descriptor to steer the curve.

2.2 Active Contours Model

In the case of the original active contour model, [Kas88], the curve γ follows a parametric representation. The curve is continuous in the image domain Ω and is represented in memory as coordinates points (See Figure 2.4).

The internal energy is defined as

$$E_{int} = \frac{1}{2}\omega_1\|\dot{\gamma}(s)\|^2 + \omega_2\|\ddot{\gamma}(s)\|^2. \tag{2.5}$$

Here, the derivatives of the curve with respect to t represent the velocity and acceleration of the points on the curve, respectively. They measure the curve length and its curvature. The

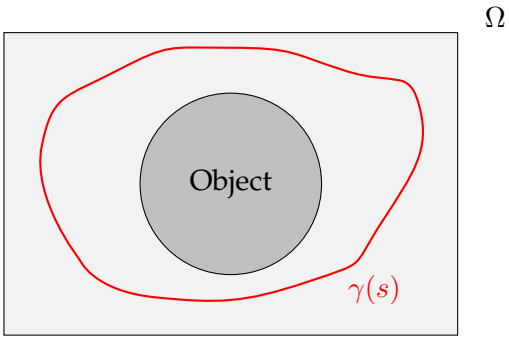


Figure 2.1: A parametric active contour $\gamma(s)$ in the image domain Ω.

coefficient ω_1, ω_2 are considered constant. And the external energy is defined as a potential P on the image domain Ω

$$P : \Omega \to \mathbb{R} \tag{2.6}$$
$$(x, y) \mapsto P(x, y), \tag{2.7}$$

To make the snake close to the boundaries, the external energy is represented as

$$E_{ext} = \lambda \int_0^1 P(\boldsymbol{\gamma}(s)), \tag{2.8}$$

where, for example,

$$P(x, y) = \|(\frac{\partial I}{\partial x}(x, y), \frac{\partial I}{\partial y}(x, y))\|^2, \tag{2.9}$$

The snake functional energy is defined over the contour $\boldsymbol{\gamma}$ as :

$$E(\boldsymbol{\gamma}) = \omega_1 \int_0^1 \|\dot{\boldsymbol{\gamma}}(s)\|^2 \, \mathrm{d}s + \omega_2 \int_0^1 \|\ddot{\boldsymbol{\gamma}}(s)\| \, \mathrm{d}s - \lambda \int_0^1 P(\gamma(s)) \, \mathrm{d}s \tag{2.10}$$

The gradient of the image is a vector that points in the direction of the steepest ascent of the image intensity, and its magnitude is the rate of this ascent. Therefore, the potential is high at the boundaries of the objects, where the image intensity changes rapidly, and it is low inside and outside the objects, where the image intensity is relatively constant. However, one significant limitation of this model is its data terms. It presumes that the contrast within the image remains constant throughout the entire area of interest or domain Ω.

This particular formulation of the model has two primary limitations. Firstly, the functional depends on the curve's parameterisation, which means the mathematical representation of the curve dramatically affects the outcome. Secondly, the topology of the curve cannot change during its evolution (See Figure 2.3b). This poses a problem when there are multiple objects in the image, and as a single contour, it cannot segment all of them.

While the curve parameterisation problem can be somewhat managed using different function bases like B-splines, tackling the topology change is a more intricate challenge. Any at-

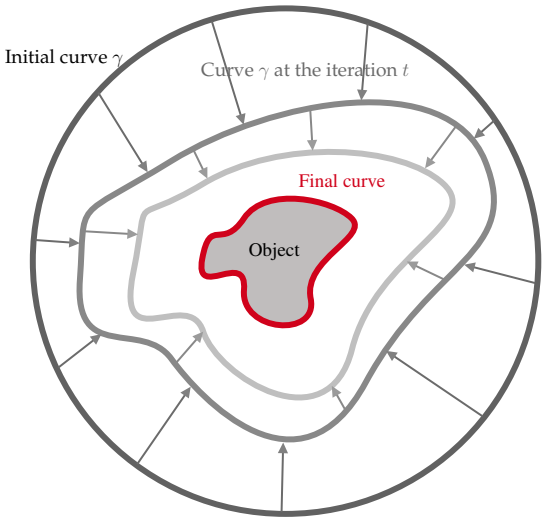
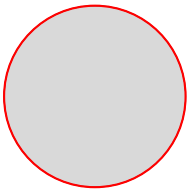
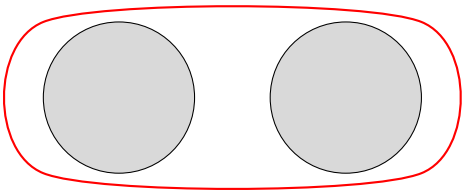


Figure 2.2: Propagation of the curve γ at various iterations t towards the object's boundaries.



(a) Active contour successfully segmenting a single object.



(b) Active contour failing to segment multiple objects due to fixed topology.

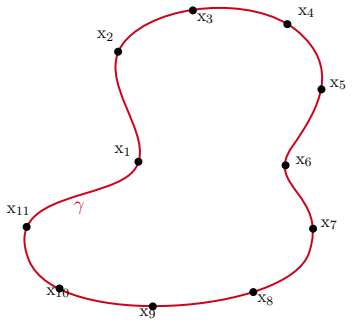


Figure 2.4: Representation of the curve γ or snake requiring discretisation of the curve.

tempt to adjust the topology during the curve's evolution would require a highly complex implementation and be computationally costly, making it an obstacle to overcome ([McI00]).

Going back to the problem at hand, we can rewrite it as

$$\mathcal{J} = \int_0^1 F(s, \boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}, \ddot{\boldsymbol{\gamma}}), \tag{2.11}$$

and the Euler-Lagrange equation of the functional $\mathcal{J}$ is expressed as

$$\frac{\partial F}{\partial \boldsymbol{\gamma}} - \frac{\partial}{\partial s}\frac{\partial F}{\partial \dot{\boldsymbol{\gamma}}} + \frac{\partial^2}{\partial s^2}\frac{\partial^2 F}{\partial \ddot{\boldsymbol{\gamma}}} = 0, \tag{2.12}$$

which, by analogy, writes

$$\nabla P(\boldsymbol{\gamma}(s)) + \omega_2 \boldsymbol{\gamma}^{(4)}(s) - \omega_1 \ddot{\boldsymbol{\gamma}}(s) = 0, \forall s \in [0,1], \tag{2.13}$$

Equation 2.13 provides us with the gradient evolution equation for the curve $\boldsymbol{\gamma}(t)$

$$\partial_t \boldsymbol{\gamma}(s) = -\omega_2 \boldsymbol{\gamma}^{(4)} + \omega_1 \ddot{\boldsymbol{\gamma}}(s) - \nabla P(\boldsymbol{\gamma}(s)). \tag{2.14}$$

This partial differential equation describes how the curve $\boldsymbol{\gamma}(s)$ evolves over time (See Figure 2.2). To solve this equation, we discretise the curve, representing it as a sequence of coordinates $\boldsymbol{\gamma} = ((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N))$. We then use an explicit scheme to approximate the solution. The second and fourth derivatives in the equation can be approximated using finite differences. Using Euler methods, we have:

$$\partial_s^2 \boldsymbol{\gamma} \simeq \boldsymbol{\gamma}_{k+1} - 2\boldsymbol{\gamma}_k + \boldsymbol{\gamma}_{k-1}, \tag{2.15}$$

$$\partial_s^4 \boldsymbol{\gamma} \simeq \boldsymbol{\gamma}_{k+2} - 4\boldsymbol{\gamma}_{k+1} + 6\boldsymbol{\gamma}_k - 4\boldsymbol{\gamma}_{k-1} + \boldsymbol{\gamma}_{k-2}. \tag{2.16}$$

where $\boldsymbol{\gamma}_k$ represents the k-th point on the curve, with coordinates (x_k, y_k).

We introduce the matrix $A = \omega_1 A_2 + \omega_2 A_4$, where A_2 and A_4 are the tridiagonal matrix and pentadiagonal matrix obtained from the second and fourth derivatives, respectively.

The tridiagonal matrix A_2 has non-zero elements on the main diagonal and the diagonals immediately above and below it:

For all k :

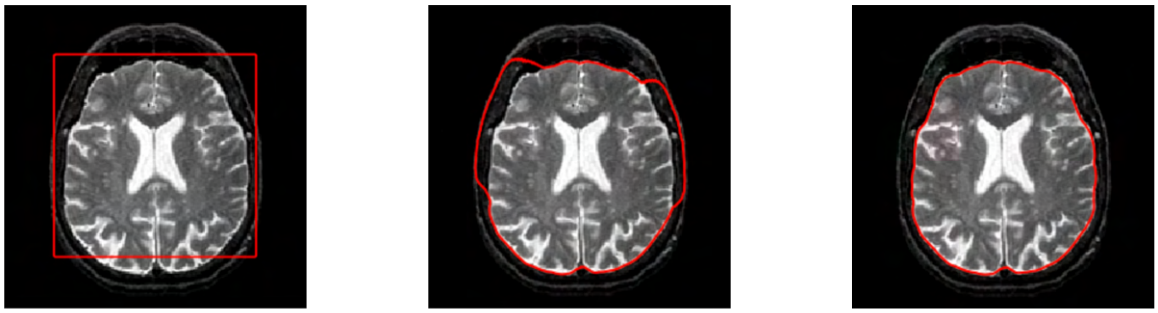


Figure 2.5: Evolution of an active contour.

$$\begin{cases} A_2[k, k-1] = 1 & \text{if } k > 1 \\ A_2[k, k] = -2 \\ A_2[k, k+1] = 1 & \text{if } k < N \end{cases}$$ The matrix A_2 captures the second derivative relationships between each point and its immediate neighbours.

The pentadiagonal matrix A_4 extends to two neighbours on each side:

For each row k :

$$\begin{cases} A_4[k, k-2] = 1 & \text{if } k > 2 \\ A_4[k, k-1] = -4 & \text{if } k > 1 \\ A_4[k, k] = 6 \\ A_4[k, k+1] = -4 & \text{if } k < N \\ A_4[k, k+2] = 1 & \text{if } k < N-1 \end{cases}$$ This matrix accounts for the fourth derivative, involving points up to two positions away.

The matrix A depends on the coefficient ω_1 and ω_2. We can then rewrite equation 2.13 in matrix form as $A \cdot \Gamma - \nabla P(\Gamma) = 0$. To solve this equation, we convert it into a fixed-point problem and use an Euler discretisation scheme with a step size (δt). This leads to the discretisation of equation 2.13:

(2.17)

This equation provides a numerical method for updating the coordinates of the curve Γ at each time step, taking into account both the internal and external forces acting on the curve.

The termination of the active contour methods can be based on the L^2 difference between two iterations and stopped when falling below a specified tolerance (See Figure 2.5).

2.2.1 Balloon Extrinsic Criterion

Cohen [Coh91] focused their work on the external criterion of data attachment, with a particular interest in the potential P from 2.10. After derivation, they defined a force as $F(v) =$

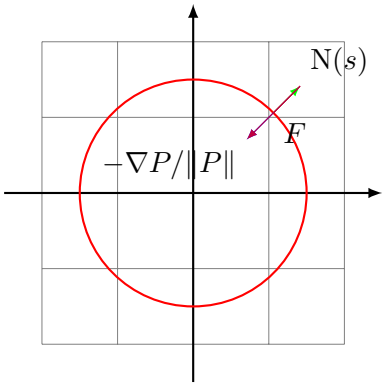


Figure 2.6: Representation of the direction of the balloon extrinsic criterion.

$-\nabla P(v)$. However, they noticed a set of challenges.

In the presence of a weak gradient, where the curve resides, it doesn't affect its evolution as $F(v)$ becomes significant compared to internal forces. Similarly, they pointed out that if the active contour is located too far from a boundary, it's not attracted by it. Without any forces tying it to the data, the curve shrinks until it disappears. This is also true for noisy images where specific standalone pixels with a maximum force $F(v)$ occasionally block the contour. They also warned that the $\frac{F(v)}{\lambda}$ ratio should not be excessively high; otherwise, point v might traverse through specific contours and start to oscillate without stabilising.

In response to these limitations, they proposed a new force for data attachment:

$$F = k_1 \mathrm{N}(s) - k\frac{\nabla P}{\|P\|}. \tag{2.18}$$

Here, $\mathrm{N}(s)$ represents the unit normal vector to the curve at a point $\boldsymbol{\gamma}(s)$. The potential P, associated with the data attachment criterion, is expected to grow as the curve moves over strong gradient regions. The constant k_1 and k serve to balance each term. The component $k\frac{\nabla P}{\|P\|}$ normalises the potential P's influence (See Figure 2.6).

The term $k_1\mathrm{N}(s)$ acts as an additional external force to excite the contour. It's like treating the snake as a balloon, and $k_1\mathrm{N}(s)$ simulates the pressure inside as it inflates. The constant k_1 dictates the magnitude of this new inflation force and, by its sign, the expansion or contraction of the evolution of the contour.

This mechanism enables the contour to move in zero-potential areas and also ensures the curves don't get trapped. However, this new force introduced by Cohen [Coh91] has a downside. It can cause the contour to change size significantly, as it can start further from the object. This necessitates re-parameterising the curve and a new matrix inversion to solve equation 2.17.

2.2.2 Geodesic Active Contour

The Geodesic Active Contour model, introduced by Caselles et al. [Cas97], was designed to reconcile Snakes with geometric definitions of contours. It integrates the geometric definition of contours with the principles of Snakes models. It is one of the first models where an energy

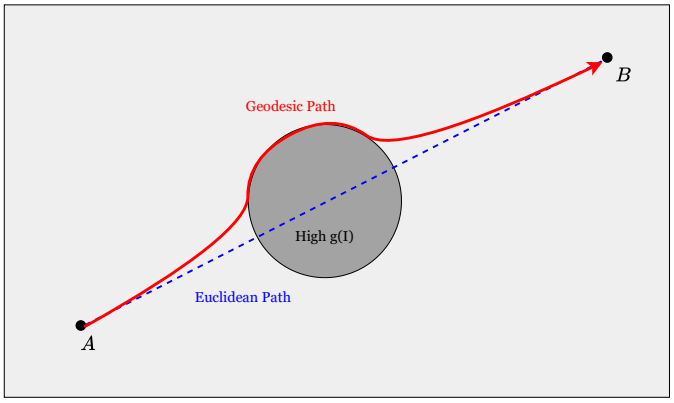


Figure 2.7: Comparison between the Euclidean shortest path (blue dashed line) and the geodesic path (red solid curve) in the image domain influenced by the metric $g(I)$. The geodesic path avoids the high-cost area represented by the circle.

functional, defined intrinsically, is implemented using level sets. This approach contrasts with previous models that primarily relied on geometric approaches.

The GAC model employs a strictly decreasing function denoted as $g : [0, \infty[\to \mathbb{R}^+$, which tends to zero as its argument approaches infinity. It is a weighting function that depends on the image intensity $I(x, y)$ (see Figure 2.7)and is used to restrict the g values and possible gaps in the boundary so that the propagating curve is guaranteed to stop. Caselles et al. [Cas97] give, for example, the function :

$$g(I) = \frac{1}{1 + |\nabla \hat{I}|^p}, \tag{2.19}$$

where $\hat{I}$ is a smoothed version of I and $p = 1$ or 2.

An energy functional, guiding the evolution of the contour, is defined as:

$$E(\boldsymbol{\gamma}(s,t)) = \omega_1 \int_0^1 \|\dot{\boldsymbol{\gamma}}\|^2 \, \mathrm{d}s + \lambda \int_0^1 g(\|\nabla \hat{I}(\boldsymbol{\gamma})\|) \, \mathrm{d}s. \tag{2.20}$$

In the initial formulation of the Geodesic Active Contour model, a notable constraint is the functional's dependence on the curve's parameterisation. This dependence introduces a level of specificity and rigidity that can reduce the model's overall applicability.

To rectify this, the original authors employed a concept from variational calculus called the Maupertuis principle. Originating from the realm of classical mechanics, the Maupertuis principle is a principle of least action, asserting that the path followed by a physical system minimises a certain quantity known as the "action".

By applying this principle, the authors demonstrate that minimising the original functional can be equivalent to determining a geodesic curve in a Riemannian space derived from the image Ω. A geodesic curve within a given space signifies the shortest possible path between two points in differential geometry.

This translation of the problem into the language of geodesics in a Riemannian space mitigates the original issue of parameterisation dependency. Instead of directly manipulating the original curve, the energy minimisation problem is transformed into finding the optimal path within a geometrically defined space. Consequently, this widens the functional's applicability and enhances its versatility in handling various image processing tasks. It was demonstrated that minimising this energy function is analogous to minimising the following equation:

$$E(\boldsymbol{\gamma}(s,t)) = \int_0^1 g(\|\nabla I(\boldsymbol{\gamma}(s,t))\|)\|\dot{\boldsymbol{\gamma}}(s,t)\|\,\mathrm{d}s. \tag{2.21}$$

Minimisation of this new functional is performed using gradient descent methods. This involves a step-by-step deformation of the curve according to the Euler-Lagrange equations :

$$\partial_t \boldsymbol{\gamma}(s,t) = (g(I(s))\kappa - \nabla g \cdot \overrightarrow{n}(s,t))\overrightarrow{n}(s,t). \tag{2.22}$$

where $\kappa = \frac{\nabla\gamma}{\|\nabla\gamma\|}$ is the curvature of the model.

This approach has the advantage of removing the need to adjust many parameters used in earlier models. The model not only halts the contour in areas with sharp changes in intensity, represented by the ∇g term, but it also attracts it to these regions.

2.2.3 Edge-based external forces

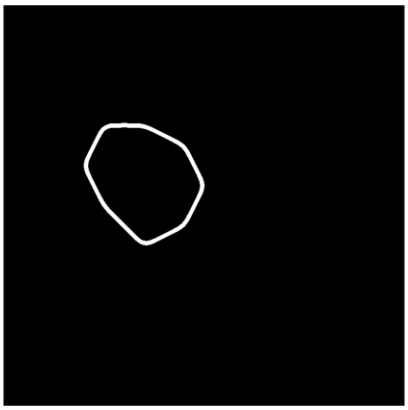
Active contour models offer a powerful means of image feature extraction through energy-minimising splines guided by external constraint forces and influenced by image forces. However, these models are often prone to getting trapped in local minima, demanding close initial positioning to the target object, and may need help with complex or poorly defined edges. To address these limitations, an improvement to the snake model incorporating local edge detection and potential functions has been proposed by Cohen et al. [Coh93b]. This method commences with edge detection using the Canny [Can86] edge extractor or other similar local edge detection techniques, providing superior starting points for contour evolution.

Upon detection of edges, a potential function is established, which derives from the distance to the closest edge and subsequently engenders an attraction force. This force draws the snake towards the detected edges, incorporating this enhancement into the original active contour model.

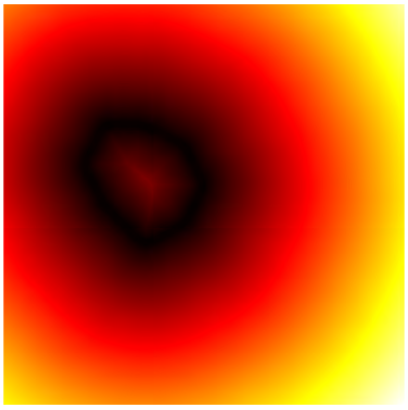
This method firstly computes an Euclidean distance map d for each point $x \in \Omega$ where $d(x)$ denotes the Euclidean distance value of x to the nearest edge points. The choice of potential function, essentially a transformation of the distance to the nearest edge, significantly influences the model's performance. Different potential functions, such as $P(v) = -e^{-d(v)^2}$ and $P(v) = \frac{-1}{d(v)}$, can modify the rate of decay of the force with increasing distance from the edge, providing granular control over the contour's evolution. In the case where $P(v)$ is defined as $g(d(v))$, the force becomes $F(v) = -\nabla P(v) = -g'(d(v))\nabla d(v)$.

The force coming from the potential function can be normalised, which makes it independent of the particular transformation function chosen. However, due to computational accuracy, the precise outcomes may still vary. If not normalised, the force can be changed based on how close or far the contour is from the edge. The process is summarised in Figure 2.8.

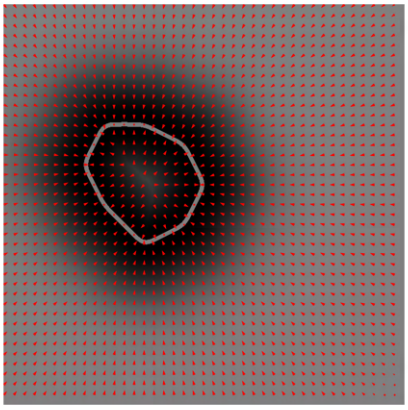
(a) The initial grayscale image containing the object of interest.



(b) Edges detected using the Canny edge detector.



(c) Distance Map $d(x)$. A heatmap representing the Euclidean distance from each pixel to the nearest edge, illustrating how distance values increase away from the edges.



(d) Visualization of the potential function derived from the distance map, with red arrows indicating the attraction forces $F(v) = -\nabla P(v)$ directing towards the edges.

Figure 2.8: Visualization of Edge-Based External Forces in Active Contour Models.

Finally, these attraction forces can be used individually or in association with an image's intensity gradient to strengthen the detected edges. This integration is particularly advantageous when dealing with fragmented detected edges.

By introducing local edge detection and using a potential function, the improved active contour model enhances the quality of segmentation and allows for improved handling of complex shapes and edges while reducing the dependence on the contour's precise initial positioning. This marks a significant stride forward in the evolution of active contour models.

2.2.4 Gradient Vector Flow

Following the idea of Cohen et al. [Coh93b], Xu et al. [Xu98] proposed a new external force based on the *Gradient Vector Flow* (GVF). Both these forces originate from an edge map of the image and can provide an extensive capture range. Studying the distance potential force introduced by Cohen et al. [Coh93b], Xu et al. [Xu98] realised that this adjustment only changes the strengths of these forces and not their direction. Consequently, the issue of the model converging to boundary concavities, or indents, remains unsolved by distance potential forces.

The *Gradient Vector Flow* (GVF) field, represented as $v(x, y) = (u(x, y), v(x, y))$, is characterised as the vector field that minimises an energy functional, denoted as:

$$E = \int\int (u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f^2| \cdot |v - \nabla f|^2 \, \mathrm{d}x \, \mathrm{d}y, \tag{2.23}$$

where ∇f represent $\|I(\cdot)\|$.

This energy functional configuration helps produce smooth results when data is absent. If ∇f is small, the energy is chiefly dictated by the aggregate of the squares of the vector field's partial derivatives, thereby providing a field of gradual variance. In contrast, when ∇f is substantial, the second term in the functional holds dominance, which is minimised when $v = \nabla f$. The effect is that v mimics the edge map gradient when substantial while maintaining slow variation in homogeneous regions.

The trade-off between the first and second terms in the integrand is managed by a regularisation parameter, μ. The parameter's value should correspond to the noise level within the image, with increased noise necessitating a higher parameter value.

Using the calculus of variations, we can validate that the GVF field is attained by solving the Euler-Lagrange equation, where ∇^2 is the Laplacian operator, and computing a gradient descent.

$$\begin{cases} \frac{\partial u}{\partial \tau} = \mu \nabla^2 u(\mathbf{x}) - (u(\mathbf{x}) - h_x(\mathbf{x})) \left\|\nabla h(\mathbf{x})\right\|^2, \\ \frac{\partial v}{\partial \tau} = \mu \nabla^2 v(\mathbf{x}) - (v(\mathbf{x}) - h_y(\mathbf{x})) \left\|\nabla h(\mathbf{x})\right\|^2, \end{cases} \tag{2.24}$$

where $h_x = \frac{\partial I}{\partial x}$.

When $I(x, y)$ is constant, the second part of each equation turns to zero because there's no gradient in f (x; y). In these areas, u and v are calculated using Laplace's equation. The resulting GVF field is then made by blending values from the region's boundary. This helps us understand why GVF fields make vectors that point into dips or hollows in the boundary. It is

like the boundary vectors competing with each other.

2.3 Level Set Method

This section presents an in-depth exploration of Level Set Methods, a mathematical technique that has significantly influenced the landscape of shape modelling and evolution. The Level Set Method, first introduced by Osher et al. [Osh88] in 1988, has since been established as a fundamental tool in computer vision, image processing, and computational physics, offering a robust and sophisticated approach to modelling the progression of shapes.

This section starts with a theoretical exposition of the Level Set Method, dissecting its mathematical foundations. The discussion will focus on the implicit representation of shapes as the zero-level set of a higher-dimensional function and how this approach facilitates the seamless handling of topological transformations. This feature proves vital in tasks such as image segmentation.

Subsequently, we will shift to the Chan-Vese model, a specific instantiation of the Level Set Method. This model, renowned for its application in image segmentation, capitalises on the strengths of the Level Set Method to segment objects without the prerequisite of edge detection. We will examine the mathematical formulation of the Chan-Vese model and discuss its practical applications.

The final part of this chapter will revisit the Geodesic Active Contour model, this time from the perspective of Level Set Methods. While we have previously explored the model's general principles and applications, this section will delve into its level-set formulation. This formulation brings to the model an inherent flexibility in handling complex shape evolutions, a characteristic central to the level-set method. We will dissect the mathematical underpinnings of this formulation, shedding light on how it leverages the strengths of level-set methods to enhance the model's performance in tasks such as image segmentation.

2.3.1 Model Definition

The level set method is a numerical method to represent and compute the evolution of a moving interface. It is based on the interface's representation as the zero-level set of a higher dimensional function. Osher et al. [Osh88] first introduced the level set method to solve the motion of a front in a fluid. Since then, it has been used in many applications such as image segmentation by Malladi et al. [Mal95] and shape optimisation from Allaire et al. [All02].

The fundamental concept revolves around the perception of a curve as a zero level-set of a function ϕ defined over the entire domain Ω (See Figure 2.9). It is the intersection between a horizontal plane to the coordinate plane and a level set. Subsequently, the curve's evolution is supplanted by a corresponding evolution of the level-set function. While this substitution might introduce increased computation by amplifying the problem's dimensionality, it also provides certain benefits. Indeed, when embedded within a level-set function, the curve is characterised by an implicit and intrinsic framework.

To better understand this, let's consider a two-dimensional image segmentation scenario. Our level set function represented as $\phi : \mathbb{R}^2 \to \mathbb{R}$, extends into a three-dimensional function.

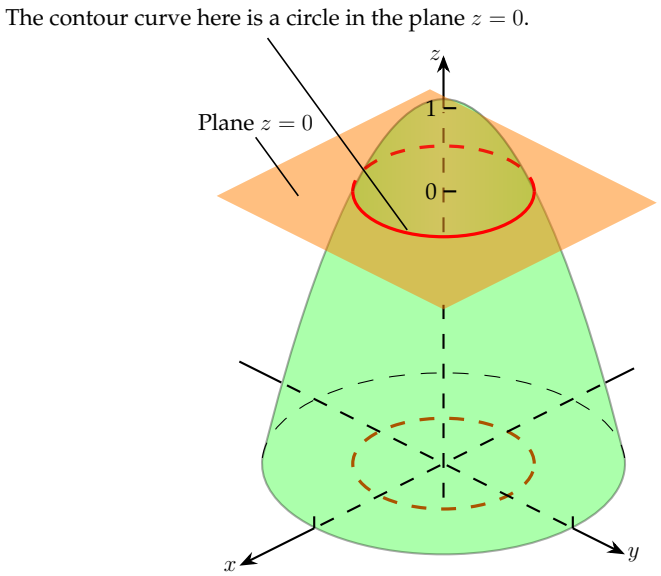


Figure 2.9: Level-set representation of the curve $\boldsymbol{\gamma}$ depicted in red, with the level-set function illustrated in green. The plane $z = 0$ denotes the zero level-set, indicating the location of the curve $\boldsymbol{\gamma}$.

This additional dimension accounts for a range of plane levels. If you imagine the function as a landscape, the desired contour would be found where this landscape intersects with a plane set at a certain level, precisely when $\phi = 0$.

If we want to define the initial contour, labelled as $\boldsymbol{\gamma}(s, 0)$, we will do so using the initial zero level of $\phi(\mathbf{x}, 0)$. Maintaining this relationship throughout the process, the contour at any given time can be represented by:

$$\forall \mathbf{x} \in \mathbb{R}^2, \forall s \in [0, 1], \boldsymbol{\gamma}(s, t) = \mathbf{x} | \phi(\mathbf{x}, t) = 0. \tag{2.25}$$

This equation essentially states that for any point $\mathbf{x}$ belonging to the contour $\boldsymbol{\gamma}(s, t)$, defined along the curvilinear abscissa s, the function $\phi(\boldsymbol{\gamma}(\mathbf{x}, t), t)$ will equal zero. As the Eulerian approach is applied, the function ϕ rather than the contour $\boldsymbol{\gamma}$ undergoes deformation. Consequently, changes in the contour $\boldsymbol{\gamma}$ are reflected directly through alterations in the zero level set of ϕ.

This form of active contours representation, often called the implicit or Eulerian representation, contrasts with the parametric representation, which transforms a model within a space to attain a final form. On the other hand, the Eulerian methodology focuses on reshaping the entire space rather than the curve itself. The changes to the model are then implicitly inferred from the overall space transformations.

Before delving deeper into the specifics of the implicit representation, it is essential to articulate how an active contour is represented in this context. Contrary to the parametric representation where an energy functional, such as equation 2.4, is sufficient to deform a contour, an implicitly represented contour demands an evolution equation. This equation delineates the deformations experienced by the curve $\boldsymbol{\gamma}(s, t)$ in its normal and tangential directions when its

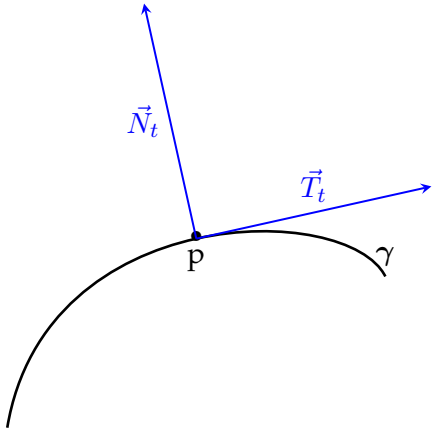


Figure 2.10: The diagram illustrates the decomposition of the curve's evolution into tangent ($\vec{T}_t$) and normal ($\vec{N}_t$) components at point $\mathbf{p}$ on the curve γ. The normal vector $\vec{N}_t$ represents the direction perpendicular to the curve, which drives the geometric evolution of the contour, either pushing it inward or outward.

energy $E(\boldsymbol{\gamma}(s,t))$ is minimised. The equation can be represented as:

$$\partial_t \boldsymbol{\gamma}(s,t) = A(s,t)T_t(\boldsymbol{\gamma}(s,t)) + B(s,t)N_t(\boldsymbol{\gamma}(s,t)), \tag{2.26}$$

where we define as $T_t(p)$ and $N_t(p)$ the unit tangent and normals at some point p at time t. And $A(\cdot,\cdot)$, $B(\cdot,\cdot)$ two scalar functions depending on the curve at time t.

The breakdown of the curve's evolution into tangent and normal components is essential, as these two aspects relate to distinct properties of the curve's evolution process. Specifically, the normal term is directly connected with the curve's geometric evolution. To understand this better, consider a simple example: imagine the contour of an object in an image, which we want to evolve or modify. The perpendicular (or normal) direction to the contour at any given point can push the contour towards the interior or the object's exterior (See Figure 2.10). This movement leads to the geometric evolution of the curve. This relation between the normal term and the geometric evolution of the curve is well explained in the subsequent property [Eps87], where a time-dependent change of parameter on the curve $\boldsymbol{\gamma}$ shows that the evolution of the curve is mainly captured by $B(\cdot,\cdot)$ and represent the speed of the normal evolution of $\boldsymbol{\gamma}$.

$$\partial_t \boldsymbol{\gamma}(s,t) = B(s,t)N_t(\boldsymbol{\gamma}(s,t)), \tag{2.27}$$

The evolution equation 2.27 defines the level sets function's deformation. Considering $\phi(s,t)$ such that its zero level set at fixed time t is the curve $\boldsymbol{\gamma}$. Then we have the equation $\phi(\boldsymbol{\gamma}(s,t),t) = 0$ for all s and t. The time derivative of this equation yields

$$\partial_t \phi(\boldsymbol{\gamma}(s,t),t)) + \nabla\phi(\boldsymbol{\gamma}(s,t),t)^T \partial_t \boldsymbol{\gamma}(s,t) = 0, \tag{2.28}$$

Then using equation 2.27 we have that

$$\partial_t \phi(\boldsymbol{\gamma}(s,t),t) = -\nabla\phi(\boldsymbol{\gamma}(s,t),t) \cdot B(s,t) N_t(\boldsymbol{\gamma}(s,t)), \tag{2.29}$$

This yields, using $N_t = -\frac{\nabla\phi(\cdot,t)}{\|\nabla\phi(\cdot,t)\|}$:

$$\partial_t \phi(\boldsymbol{\gamma}(s,t),t) = B(s,t) \left\|\nabla\phi(\boldsymbol{\gamma}(s,t),t)\right\|. \tag{2.30}$$

Finally, if we can define $B(\cdot,\cdot)$ on the all domaine Ω then we can introduce the global evolution equation

$$\partial_t \phi(p,t) = B(p,t) \left\|\nabla\phi(p,t)\right\|, \forall p \in \Omega. \tag{2.31}$$

The principal advantages of the level-set method are its robustness, adaptability, and potency. It is an excellent approach to address problems related to the evolution of shapes, particularly image segmentation and shape modelling tasks. This technique leverages the implicit definition of contours through a function of dimension $(n+1)$. It details their evolution by employing a partial differential equation, thus facilitating the manipulation of contours and managing topological transformations, including splits and merges.

Using a speed function in the equation adds flexibility to the method. It lets us include specific details of the problem we're working on, like guiding the contour towards image features. Also, the method considers changes in the whole computational area, not just the contour itself, giving a more complete view of the changes in the contours.

As we will see in the next paragraph, the level-set method can require a lot of computing power. This is because it needs to solve equations across a grid that covers the whole area, no matter how complex the contours are. So, while it's a powerful method, it's also essential to consider its high computational demand.

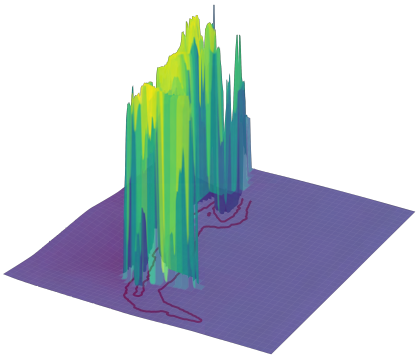
2.3.2 Chan-Vese Model

The region-based active contour model developed by Chan et al. [Cha01] in 2001 is seen as a particular instance of the Mumford-Shah model from 1989 [Mum89]. This model proposes an energy minimisation problem to find the best partition of an image into two distinct regions delineated by a contour $\boldsymbol{\gamma}$. Instead of limiting the analysis to areas near high gradient lines on the image, it can focus on differences in image properties within and outside the shape. This gives rise to the Chan-Vese variational problem, which strives to minimise an energy function that adjusts contours such that the image intensity remains nearly constant within and outside these contours (See Figure 2.11).

To elaborate, let's consider $\Omega \in \mathbb{R}^2$ as the image's domain (a square or a rectangle) and $\mathrm{I} : \Omega \to \mathbb{R}$ as the image function itself. Rather than directly defining a curve in Ω to denote the shape, we consider an open subset M of Ω, which signifies the shape's interior.

The assumption is that image I display almost constant grey levels on both M and the complement of M (M^c), albeit with a potential discontinuity at the boundary ∂M. The energy function will enforce this condition, along with ensuring the smoothness of the contour, ∂M. This is achieved by controlling both its length and the area within it.

The model introduces two parameters: c_1, representing the mean grey level intensity within



(a) Level Set Function $\phi(x, y)$

(b) Segmentation Boundary (Zero Level Set) Overlaid on the Original Image

Figure 2.11: Visualization of the Chan-Vese Segmentation Process.

the region enclosed by contour ∂M, and c_2, its counterpart for the region external to the contour. To ensure a smooth and regular contour, intrinsic constraints are incorporated, such as the length of contour ∂M and its enclosed area.

The Chan-Vese Energy is defined as

$$E(c_1, c_2, M) = \mu \times \text{Length}(\partial M) + \nu \times \text{Area}(M) \tag{2.32}$$

$$+\lambda_1 \int_M \|\mathrm{I}(x) - c_1\| \, \mathrm{d}x + \lambda_2 \int_{M^c} \|\mathrm{I}(x) - c_2\| \, \mathrm{d}x \tag{2.33}$$

The energy described is essentially a simplified version of the Mumford-Shah functional [Mum89] formulated to approximate an observed image with a piecewise smooth function. The Chan-Vese model can be seen as a piecewise-constant approximation of the Mumford-Shah functional.

Implementing the minimisation of this energy function can be intricate. This is primarily due to the energy function's dependency on the set M, as opposed to a function that uses a more conventional approach in the calculus of variation. To circumvent this problem, the author introduces an implicit representation of ∂M as the zero level set of some ϕ function such that for all p in ∂M, $\phi(p) = 0$. We suppose that

$$\begin{cases} f(p) = 0 \text{ if } p \in \partial M \\ f(p) < 0 \text{ if } p \in M \\ f(p) > 0 \text{ if } p \in M^c \end{cases} \tag{2.34}$$

Using the Heaviside function and the Dirac measure defined as:

$$H(z) = \begin{cases} 1 \text{ if } z \geq 0 \\ 0 \text{ if } z < 0 \end{cases}, \delta_0(z) = \frac{d}{dx} H(z) \tag{2.35}$$

They also introduce the smooth approximation of the Heaviside function defined on $\mathbb{R}$ as H_0 by

$$H_\epsilon(x) = \frac{1}{2}(1 + \frac{2}{\pi} \arctan(\frac{x}{\epsilon})). \tag{2.36}$$

The derivative of the Heaviside function is defined by the Dirac function δ_0 and, in this case, the approximation of the Dirac function δ_ϵ. Then, one can write:

$$\begin{aligned} E(c_1, c_2, \phi) =& \mu \int_\Omega \delta_\epsilon(\phi(x)) \left\| \nabla \phi(x) \right\| \, \mathrm{d}x + \nu \int_\Omega H_\epsilon(\phi(x)) \, \mathrm{d}x \\ &+ \lambda_1 \int_\Omega \left\| \mathrm{I}(x) - c_1 \right\|^2 H_\epsilon(\phi(x)) \, \mathrm{d}x + \nu \int_\Omega \left\| \mathrm{I}(x) - c_2 \right\|^2 (1 - H_\epsilon(\phi(x))) \, \mathrm{d}x. \end{aligned} \tag{2.37}$$

We can express the c_1 and c_2 exactly

$$\begin{cases} c_1 = \dfrac{\int_\Omega H_\epsilon(\phi(x)) \mathrm{I}(x) \, \mathrm{d}x}{\int_\Omega H_\epsilon(\phi(x)) \, \mathrm{d}x} \\ \\ c_2 = \dfrac{\int_\Omega (1 - H_\epsilon(\phi(x))) \mathrm{I}(x) \, \mathrm{d}x}{\int_\Omega (1 - H_\epsilon(\phi(x))) \, \mathrm{d}x}. \end{cases} \tag{2.38}$$

To apply a gradient descent algorithm, we must first find the associated Euler-Lagrange equation of the energy functional.

The Euler-Lagrange equation is as

$$\partial_t E(c_1, c_2, \phi) = \int_\Omega h(x) \delta_\epsilon(\phi(x)) (-\mu \operatorname{div}(\frac{\nabla \phi}{\left\| \nabla \phi \right\|}) - \nu + \lambda_1 \left\| \mathrm{I}(x) - c_1 \right\|^2 - \lambda_2 \left\| \mathrm{I}(x) - c_2 \right\|^2) \, \mathrm{d}x. \tag{2.39}$$

The different terms of the energy can be interpreted as follows:

- The first term is the regularisation term. It ensures that the level set function remains a signed distance function to the contour.
- The second term is the data term. It ensures that the mean intensity of the region inside the contour is close to c_1.
- The third term is the data term. It ensures that the mean intensity of the region outside the contour is close to c_2.

This is important because the level set method can segment an image into two regions with different mean intensities. The level set method can also segment an image into more than two

regions.

2.3.2.1 Numerical implementation

This section details the semi-implicit gradient descent method for solving the Chan–Vese minimisation problem, based on the work of Getreuer [Get12], as developed in the foundational works Chan and Vese [Cha01]. This approach is one among several for addressing the minimisation problem; alternative methods include the topological derivative algorithm by He and Osher [He07], the multigrid method by Badshah and Chen [Bad08], and fast algorithms based on graph cuts by Zehiry, Xu, and Sahoo El Zehiry, Xu, Sahoo, and Elmaghraby [EZ07], and Bae and Tai [Bae09].

For numerical implementation, consider the function f sampled on a regular grid $\Omega = \{0, \dots, M\} \times \{0, \dots, M\}$. The evolution of φ is discretised spatially according to the following equation:

$$\frac{\partial \varphi_{i,j}}{\partial t} = \delta_\epsilon(\varphi_{i,j}) \left[\mu \left(\frac{\nabla_x^- \nabla_x^+ \varphi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^+ \varphi_{i,j})^2 + (\nabla_y^0 \varphi_{i,j})^2}} + \frac{\nabla_y^- \nabla_y^+ \varphi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^0 \varphi_{i,j})^2 + (\nabla_y^+ \varphi_{i,j})^2}} \right) \right. \tag{2.40}$$

$$\left. - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \right] \tag{2.41}$$

where ∇_x^+ denotes the forward difference in the x dimension, ∇_x^- denotes the backward difference, and $\nabla_x^0 := (\nabla_x^+ + \nabla_x^-)/2$ is the central difference, with analogous definitions in the y dimension. The parameter η regularises the curvature term, preventing division by zero; typically, $\eta = 10^{-8}$.

Defining auxiliary variables:

$$A_{i,j} = \mu \frac{1}{\sqrt{\eta^2 + (\nabla_x^+ \varphi_{i,j})^2 + (\nabla_y^0 \varphi_{i,j})^2}}, \quad B_{i,j} = \mu \frac{1}{\sqrt{\eta^2 + (\nabla_x^0 \varphi_{i,j})^2 + (\nabla_y^+ \varphi_{i,j})^2}} \tag{2.42}$$

the discretised evolution equation becomes:

$$\frac{\partial \varphi_{i,j}}{\partial t} = \delta_\epsilon(\varphi_{i,j}) \left[A_{i,j}(\varphi_{i+1,j} - \varphi_{i,j}) - A_{i-1,j}(\varphi_{i,j} - \varphi_{i-1,j}) + B_{i,j}(\varphi_{i,j+1} - \varphi_{i,j}) \right. \tag{2.43}$$

$$\left. - B_{i,j-1}(\varphi_{i,j} - \varphi_{i,j-1}) - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \right] \tag{2.44}$$

The right-hand side terms discretise the curvature term $\operatorname{div}\left(\frac{\nabla \varphi}{|\nabla \varphi|}\right)$, ensuring that the mixed differences combine to produce a centred yet localised result. Specifically, the forward and backward differences are applied so that the resulting numerator and denominator are logically centred at the desired grid points, enhancing the accuracy of the curvature approximation.

Time discretisation employs a semi-implicit Gauss-Seidel method [5], allowing in-place updates of φ values, thus optimising memory usage. The update rule is:

$$\frac{\varphi_{i,j}^{n+1} - \varphi_{i,j}^{n}}{\Delta t} = \delta_\epsilon(\varphi_{i,j}^n) \Big[A_{i,j}\varphi_{i+1,j}^n + A_{i-1,j}\varphi_{i-1,j}^{n+1} + B_{i,j}\varphi_{i,j+1}^n + B_{i,j-1}\varphi_{i,j-1}^{n+1} \tag{2.45}$$

$$- (A_{i,j} + A_{i-1,j} + B_{i,j} + B_{i,j-1})\varphi_{i,j}^{n+1} - \nu - \lambda_1(f_{i,j} - c_1)^2 + \lambda_2(f_{i,j} - c_2)^2 \Big] \tag{2.46}$$

Here, $\varphi_{i,j}$, $\varphi_{i-1,j}$, and $\varphi_{i,j-1}$ are evaluated at time step $n+1$, while other terms remain at time step n. This strategy ensures stability and accuracy in the numerical solution of the Chan–Vese minimisation problem.

This allows φ at timestep $n+1$ to be solved by one Gauss-Seidel sweep from left to right, top to bottom:

$$\varphi_{i,j}^{n+1} \leftarrow \Big[\varphi_{i,j}^n + \Delta t \delta_\epsilon(\varphi_{i,j}^n) \Big[A_{i,j}\varphi_{i+1,j}^n + A_{i-1,j}\varphi_{i-1,j}^{n+1} + B_{i,j}\varphi_{i,j+1}^n + B_{i,j-1}\ \varphi_{i,j-1}^{n+1} \tag{2.47}$$

$$- \nu - \lambda_1(f_{i,j} - c_1)^2 + \lambda_2(f_{i,j} - c_2)^2 \Big] \Big] \,/\, (1 + \Delta t \delta_\epsilon(\varphi_{i,j})(A_{i,j} + A_{i-1,j} + B_{i,j} + B_{i,j-1})) \tag{2.48}$$

The coefficients A and B are computed using the latest available values of φ:

$$A_{i,j} = \mu \frac{1}{\sqrt{\eta^2 + (\varphi_{i+1,j}^n - \varphi_{i,j}^n)^2 + \left(\frac{\varphi_{i,j+1}^n - \varphi_{i,j-1}^{n+1}}{2} \right)^2}}, \tag{2.49}$$

$$B_{i,j} = \mu \frac{1}{\sqrt{\eta^2 + \left(\frac{\varphi_{i+1,j}^n - \varphi_{i-1,j}^{n+1}}{2} \right)^2 + (\varphi_{i,j}^n - \varphi_{i+1,j}^n)^2}}. \tag{2.50}$$

Boundary conditions are enforced by duplicating pixels near the borders:

$$\varphi_{-1,j} = \varphi_{0,j}, \quad \varphi_{M,j} = \varphi_{M-1,j}, \quad \varphi_{i,-1} = \varphi_{i,0}, \quad \varphi_{i,M} = \varphi_{i,M-1}. \tag{2.51}$$

Optionally, the level set function can be reinitialised after every N iteration by replacing φ with the signed distance function to C or any other function having the same sign at each point. This reinitialisation does not modify the segmentation boundary but prevents new components from appearing far away from the current boundary.

The termination of the method can be based on the L^2 difference between φ^{n+1} and φ^n falling below a specified tolerance. In the implementation, the default value of tol is 10^{-3}. The overall algorithm, as described in the original paper [9], has a linear computational cost per iteration in the number of pixels.

The required number of iterations depends significantly on the timestep Δt and the initialisation. The contour evolves slowly if it has low curvature (e.g., a large ellipse), potentially requiring thousands of iterations to converge. An initialisation with high curvature, such as the checkerboard initialisation mentioned in the previous section, tends to converge much faster.

This semi-implicit approach efficiently balances stability and computational complexity,

making it a robust choice for solving the Chan–Vese minimisation problem in various image segmentation applications.

2.4 Geodesics

From primary school to university, we are taught to compute distances in a two-dimensional plane. The fundamental principle of Euclidean geometry is that the shortest distance between two points is a straight line. This concept is grounded in our minds and becomes second nature to us. However, when we observe the world around us, particularly the trajectories of aeroplanes, we notice that they do not follow this rule.

For example, a flight from Paris to New York does not travel in a straight line. Instead, it goes up north before reaching its destination. This is because the shortest distance between two points on a sphere is not a straight line but an arc. This shortest path is a geodesic, a fundamental concept in non-Euclidean geometry.

In other words, the geometry governing our planet differs from the Euclidian geometry we learn in high school. The Earth is not a flat surface but a three-dimensional sphere, and therefore, the rules that apply to it are different. Its curvature allows us to calculate the shortest distance between two points on any surface.

Definition 1 (Geodesic Distance)**.** In a Riemannian manifold M with metric tensor g, we define the geodesic distance as the minimal length L between two endpoints a and b of a continuously differentiable curve $\gamma : [a, b] \to M$

$$L(\gamma) = \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt. \tag{2.52}$$

The metric tensor g is a symmetric, positive-definite bilinear form on the tangent space T_pM of the manifold at point p. g is a inner product between any two tangent vectors $\forall u, v \in T_pM, g_p(u, v) =< u, v >_p$.

Computing the distance between two points on a surface is a more complex problem than computing the Euclidean distance between two points in a plane. While the Euclidean distance can be easily calculated using the Pythagorean theorem, computing the distance between two points on a surface requires considering the surface's curvature and shape.

One approach to computing the distance between two points on a surface is to use geodesics, the shortest curves connecting two points on a surface. The length of a geodesic between two points is the distance between those points on the surface. However, finding geodesics on a surface can be challenging, especially for surfaces with complex shapes.

Several algorithms have been developed to compute geodesic distances on surfaces. One popular approach is the Fast Marching Method, an efficient numerical algorithm for computing the distance between a starting point and all other points on a surface. The algorithm propagates a wavefront from the starting point outward, computing the distance to neighbouring points as the wavefront advances. The algorithm terminates when the wavefront reaches all points on the surface.

2.4.1 The Eikonal Equation

In order to find the geodesic distance u it wan be shown that it satisfies a partial differential equation called the Eikonale Equation. The Eikonal Equation is defined as follows:

$$\begin{cases} a(x)^{-1}\|Du(x)\| = 1, \forall x \in \Omega \\ u = 0 \text{ on } \partial\Omega \end{cases} . \tag{2.53}$$

with Ω an open and bounded set in $\mathbb{R}^n$. The distance is not everywhere differentiable to the boundary. We have to find a viscosity solution to the Eikonal equation.

Definition 2 (Viscosity)**.** A continuous function $u : \bar{\Omega} \to \mathbb{R}$ is a viscosity sub-solution (resp. super-solution) of 2.53 if :

1. $u(x) \leq 0$ (resp. $u(x) \geq 0$) for all x in $\partial\Omega$.
2. Given $r > 0, x_0 \in \mathcal{B}(x_0, r) \subseteq \Omega$, if $\Phi : \mathcal{B}(x_0, r) \to \mathbb{R}$ is a smooth function such that $u - \Phi$ has a local maximum (resp. minimum) at x_0, then $\|D\Phi(x_0)\| \leq 1$ (resp. $\|D\Phi(x_0)\| \geq 1$).

A continuous function is a viscosity solution of 2.53 if it is both a viscosity sub- and super-solution.

2.4.2 The Fast Marching Algorithm

Older methods, such as Gauss-Seidel, are slow to compute a solution to solve the Eikonal equation because it requires several times passing through each grid point. Methods have been developed to improve the complexity in $\mathcal{O}(N^{1+1/d})$, where N is the discrete domain cardinality and d is the domain dimension, of Gauss-Seidel methods such as fast sweeping methods that alternate sweeps along the 2d directions of the grid [Zha05] or the use of a priority queue [Bor06]. The Fast Marching Method, developed by Sethian [Set96], is an efficient algorithm for computing the minimal action map or geodesic distance map for an isotropic Riemannian metric by computing exactly the solution in $O(N \log(N))$ operations, where N is the number of sampling points. The algorithm is based on a monotonically advancing wave propagation manner, similar to Dijkstra's non-iterative algorithm [DIJ59], but with a discretisation scheme for the local geodesic distance update that differs between Sethian's method and Tsitsiklis's shortest path method ([Tsi95]).

We first introduce some basic notations to estimate the minimal action map U using the fast marching method. Let Z be a discretisation orthogonal grid of the domain Ω with dimension d, and let N be the total number of grid points of Z.

The fast marching method is based on an optimal ordering of the grid points that ensures that each point is visited only once by the algorithm and that this visit computes the exact solution. The algorithm starts from the initial source points and propagates outward until filling the whole domain. During the propagation, each grid point in Z is labelled according to a state: Computed, Front, or Far. Computed points are the grid points for which minimal action values of U have been estimated and frozen; the algorithm will not consider them any more. Front points are the grid points for which the minimal action values have been calculated

but not frozen; they are the points being processed. Far points are the grid points for which the minimal action values have not been estimated. The value of the distance map for Front points is well-defined but might change in future iterations, while for Far points, it is defined as ∞.

The fast marching front is the interface between the Computed and Far points and consists of all the Front points. The Front points are stored in a priority queue such that the Front point xmin with the smallest value of U can be identified efficiently. By marching the front in an ordered way, the minimal action map U can be obtained within a finite number of local geodesic distance update steps.

The overview of the fast marching method is presented in Algorithm 1. In each step, the grid point $x_{\min}$ with the smallest value of U among all the Front points is selected and tagged as Computed. The neighbourhood points y of xmin are then updated by the local geodesic distance update scheme detailed in Sections 2.4.2, 2.4.3, and 2.4.5. The stopping criterion for the fast marching algorithm is when all the grid points in Z have been tagged as Computed. However, an early abort scheme can be applied to reduce the computation time: once all the endpoints are tagged as Computed, the fast marching can be stopped. Figure 2.12 presents the computation of the distance using the Fast Marching algorithm in a maze.

Algorithm 1 An algorithm with caption

Require: $s_i, i \in \{1, ..., m\}$ The initial source points, $\mathcal{F}$ a metric

```
Create a priority queue P and add the starting point with a distance d of 0
while the P is not empty do
    Remove the point with the smallest distance from the queue
    for each neighbour of the removed point do
        if the neighbour is not known then
            Compute d_trial to the neighbour using the local geodesic distance update scheme
            if d_trial is smaller than d then
                Update the distance of the neighbour and its parent
                Add the neighbour to P
```

2.4.3 The Fast Marching Method on 2D Grid

Using a stationary approach, the Fast Marching algorithm aims to solve the Eikonal Equation, a non-linear Partial Derivatives Equation. The Eikonal Equation is given by:

$$|\nabla u(x)| = n(x), \tag{2.54}$$

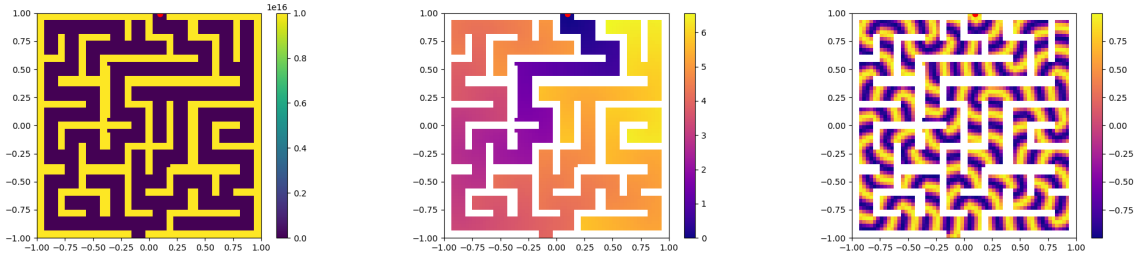
where x is in an open subset of $\mathbb{R}^n$ and $n(x)$ is a positive function. In the context of geometric optics, n is the refractive index of the medium.

Writing it as a wave equation, we have:

$$\partial_t u(x,t) + c(x)|\nabla_x u(x,t)| = 0, \tag{2.55}$$

where x represents a point in space, t is the time and $c(x)$ the function of the speed.

Let $T(x)$ be a function giving the arrival time of the front at a point x in space. We denote



(a) (b) (c)

Figure 2.12: The images illustrate the potential field and geodesic distance computation in a maze-like environment using the Fast Marching (FM) method. In (a), the potential field is visualised, which serves as the input to the FM algorithm. In (b), the geodesic distance from a source point (marked in red) to all other points in the maze is shown. Lastly, in (c), the distance is modulated with a sinusoidal function to show the level-sets.

the propagating front at time t as Γ_t. This function satisfies the following condition for all trajectories $t \to y(t) \in \Gamma_t$.

$$T(y(t)) = t. \tag{2.56}$$

By differentiating the equation with regards to t and using the expression of the normal exterior to the front Γ_t,

$$\vec{n}_{\Gamma_t} = \frac{\nabla_x T(x)}{|\nabla_x T(x)|}, \tag{2.57}$$

We get the following equation for the function T:

$$c(y(t))|\nabla_x T(y(t))| = 1. \tag{2.58}$$

This equation must be solved on the whole domain, which led to the stationary equation:

$$c(x)|\nabla_x T(x)| = 1, \tag{2.59}$$

with the boundary conditions $T(x) = 0$ for $x \in \Gamma_0$.

The goal is to numerically compute the arrival time T, which is the solution of the following stationary equation:

$$c(x)|\nabla_x T(x)| = 1. \tag{2.60}$$

We consider the 2D case with $x = (x_1, x_2)$ on a Cartesian grid with spatial step $h = (\Delta x_1, \Delta x_2)$. The proposed discretisation of the stationary equation is:

$$\max\left(\frac{T_{i,j}-T_{i-1,j}}{\Delta x_1},-\frac{T_{i+1,j}-T_{i,j}}{\Delta x_1},0\right)^2 \tag{2.61}$$

$$+\max\left(\frac{T_{i,j}-T_{i,j-1}}{\Delta x_2},-\frac{T_{i,j+1}-T_{i,j}}{\Delta x_2},0\right)^2=\frac{1}{c^2(x_{i,j})}. \tag{2.62}$$

An initial approach could be to apply an iterative scheme to solve this nonlinear problem until the algorithm converges with the desired precision. However, this method can take a lot of work. The Fast-Marching method proposes calculating the values $T_{i,j}$ in a particular order, allowing moral convergence in a single iteration by calculating $T_{i,j}$ in ascending order.

To implement this method, the grid points x_i, j are divided into three regions:

- Frozen Points: Definitively calculated points. These are the points that the front has already cut.
- Narrow Band: Points about to be cut by the front, having a neighbour already frozen.
- Far Away Points: Other points not yet reached by the front.

Initialisation:

- Initial Front: Points on the boundary are frozen, and T is initialised to 0 on these points.
- Narrow Band: Composed of the immediate neighbours of the initial boundary in the direction of front propagation.
- Far Away Points: Consists of the other points in the direction of front propagation. T is initialized to ∞ on these points.

Initial Calculation of T on the Narrow Band
The values of T on the narrow band are initialised with the following formulas:

$$\left(\frac{T_{i,j}}{\Delta x_1}\right)^2=\frac{1}{c^2(x_{i,j})}, \tag{2.63}$$

$$\left(\frac{T_{i,j}}{\Delta x_2}\right)^2=\frac{1}{c^2(x_{i,j})}, \tag{2.64}$$

$$\left(\frac{T_{i,j}}{\Delta x_1}\right)^2+\left(\frac{T_{i,j}}{\Delta x_2}\right)^2=\frac{1}{c^2(x_{i,j})}. \tag{2.65}$$

This calculation assumes that the value of T on the current point is smaller than that of the narrow band neighbours and larger than that of the neighbours on the initial front (respectively smaller for the far away region).

Fast-Marching Algorithm:

1. Find the Smallest Value of T on the Narrow Band: The corresponding point becomes accepted (frozen).

2. Redefine the Narrow Band: Add the neighbours of the newly accepted point to the narrow band.

3. Recalculate T on the Neighboring Points: Solve the discretisation of the stationary equation to update the values of T on the points X neighbouring the newly accepted point A.

2.4.4 Implemetation of the Fast-Marching Method

To implement the Fast-Marching method, we need to manipulate several arrays:

1. **Array $\mathbf{T}(\mathbf{i},\mathbf{j})$** : Contains the values $T_{i,j}$ at the nodes $x_{i,j}$.

2. **Array $\mathbf{TAB}(\mathbf{i},\mathbf{j})$** : Indicates the nature of the point $x_{i,j}$. For example, $TAB(i,j) = 1$ if the point is frozen, $TAB(i,j) = -1$ if it is in the narrow band, and $TAB(i,j) = 0$ otherwise (far away region).

3. **Array $\mathbf{Pile}(\mathbf{i},\mathbf{j},\mathbf{T}(\mathbf{i},\mathbf{j}))$** : Contains the indices of the elements in the narrow band. The rows of $Pile$ will be sorted according to the increasing values of the third column, thus requiring a sorting algorithm.

4. **Array $\mathbf{Pile\_test}(\mathbf{i},\mathbf{j})$** : Contains the indices of the four neighbouring points of a newly accepted point. These 4 points, if not already accepted, will be added to the array $Pile(i,j,T(i,j))$ (if they are not already there), and the value of $T(i,j)$ will be recalculated.

The scheme is written as follows:

$$\max\left(\frac{T_{i,j}-T_{i-1,j}}{\Delta x_1}, -\frac{T_{i+1,j}-T_{i,j}}{\Delta x_1}, 0\right)^2 + \max\left(\frac{T_{i,j}-T_{i,j-1}}{\Delta x_2}, -\frac{T_{i,j+1}-T_{i,j}}{\Delta x_2}, 0\right)^2 = \frac{1}{c^2(x_{i,j})}. \tag{2.66}$$

To update the narrow band at each iteration, we will need to solve for $T_{i,j}$, given fixed values $(t_1,t_2) := (T_{i-1,j}, T_{i+1,j})$ and $(t_3,t_4) := (T_{i,j-1}, T_{i,j+1})$. The goal is to find θ that solves:

$$\max\left(\frac{\theta-t_1}{\Delta x_1}, -\frac{t_2-\theta}{\Delta x_1}, 0\right)^2 + \max\left(\frac{\theta-t_3}{\Delta x_2}, -\frac{t_4-\theta}{\Delta x_2}, 0\right)^2 = \frac{1}{c^2}. \tag{2.67}$$

In other words:

$$\frac{1}{\Delta x_1^2}\left(\theta - \min(t_1,t_2,\theta)\right)^2 + \frac{1}{\Delta x_2^2}\left(\theta - \min(t_3,t_4,\theta)\right)^2 = \frac{1}{c^2}, \tag{2.68}$$

where, for simplicity, we denote $c = c(x_{i,j})$. Let $h_1 = \Delta x_1$, $h_2 = \Delta x_2$, $v_1 = \min(t_1,t_2)$, and $v_2 = \min(t_3,t_4)$. The equation can also be written as:

$$\frac{1}{h_1^2}\left(\theta - \min(v_1,\theta)\right)^2 + \frac{1}{h_2^2}\left(\theta - \min(v_2,\theta)\right)^2 = \frac{1}{c^2}. \tag{2.69}$$

2.5 Distance transform

A distance transform is a grey-level image in which the intensity shows the distance between the pixel to the nearest edge of a set of pixels.

One commonly used metric is the p-norm, the Minkowski distance.

Definition 3. For all x, y in $\mathbb{R}^N$, $p > 0$, the Minskowski distance is defined as:

$$d(x, y) = \|x - y\|_p = \sqrt[p]{\sum_{n=1}^{N} |x_n - y_n|^p} \tag{2.70}$$

In the case of a distance transform, the input to this distance function is a binary image, I, where each pixel can take two values, 0 and 1. 0 to be the background and 1 to be the pixels of interest.

The distance transform, D, is a grayscale image generated from the binary image, I. the value of each pixel in D corresponds to the minimum distance between that pixel's location and the surrounding edge pixels.

Definition 4. For all x, y in I, the distance transform is defined as:

$$D(x) = \min_{\{y: I(y)=1\}} d(x, y) \tag{2.71}$$

Calculating the distance transform involves up to N comparisons for each pixel, resulting in a $O(N^2)$ complexity. However, techniques to reduce this complexity can be categorised into three types: propagation, raster-scanning and separable scanning.

Propagation algorithms compute the distance transforms by progressively moving away from the edge pixels and recording the distances. Raster-scanning algorithms approximate the Euclidean distance using Chamfer distance, with local masks chosen to minimise the approximation error. Separable scanning algorithms reduce the operations into independent one-dimensional operations by tracking parabola intersections or using morphological operators.

The distance function is computationally expensive because of the min operation. This operation is highly nonlinear, which makes it challenging to accelerate. This section will see one alternative form for the minimum operation, also known as smooth operations to minimum functions. These are commonly used in machine learning algorithms. When these smooth approximations are substituted into the definition of the distance transform 2.71, the algorithm can be efficiently approximated using convolution operators. The motivation to compute the distance transform differently is also to have all operations differentiable. This approximation can then be integrated as a differentiable convolutional distance transform layer into current deep learning frameworks.

We first rewrite the minimum function using the log-sum-exponential form. We recall the following lemma due to Karam et al. [Kar19].

Lemma 1. *Let $z_1, \cdots, z_K \in \mathbb{R}$. Then,*

$$\min\{z_1, \cdots, z_K\} = \lim_{\lambda \to 0} -\lambda \log \left(\sum_{k=1}^{K} \exp\left(-\frac{z_k}{\lambda}\right) \right). \tag{2.72}$$

This equation is a way to approximate the minimum function using a smoother, differentiable function.

In this section, we will focus on translation-invariant metrics. This means that the distance dunction $d(\cdot, \cdot)$ has the following porperty : $d(x, y) = d(x + z, y + z)$ for all z in $\mathbb{R}^N$.

We recall the following theorem :

Theorem 1. *Let $*$ denote the convolution. Then*

$$D(x) = \lim_{\lambda \to 0} -\lambda \log \left(I(x) * \exp\left(-\frac{d(x)}{\lambda}\right) \right). \tag{2.73}$$

This theorem allows us to reformulate the distance transform as a convolution, which can be computed more efficiently. The distance transform can be approximated using a convolution of the binary image I with a kernel $\exp(-d(\cdot, 0)/\lambda)$, where $*$ is the convolutional operator.

One of the main challenges of the convolutional design of the distance transform is that the kernel size needs to be as large as the diagonal of the input image to ensure that even very sparse binary images can be distance transformed. However, this leads to two main issues: increased computational complexity and decreased numeric stability for vast distances. To address this issue, [Pha21] propose a cascade of local distance transforms.

The large kernel size required for the convolutional operation increases computational complexity, making it impractical for large images. Additionally, for considerable distances, the exponential term in the kernel design may approach zero, leading to decreased numeric stability of the logarithmic expression within the *Convolutional Distance Transform* (CDT). This issue is particularly noticeable in large images with only a few foreground pixels.

To tackle this challenge, [Pha21] suggests cascading distance transforms with smaller kernels to approximate the actual transform. This approach reduces the computational complexity and overcomes the numerical instability. Instead of directly computing the distance transform with a large kernel, we iteratively extend the binary input image by the area for which a distance calculation was possible by the locally restrictive CDT. The calculated distances are then used with the distances for the previous iterations to form the final distance transform.

The maximum distance to a foreground pixel that can be captured by the CDT is limited to a range of $\lfloor \frac{k}{2} \rfloor$, k the kernel size. For all background points that are further away than $\lfloor \frac{k}{2} \rfloor$ from a foreground point, 1 yields a distance of 0. The idea is to iteratively extend the binary input image by the area for which a distance calculation was possible, i.e. by all points that fulfil the condition that the calculated distance is more significant than zero. This extended binary image can then compute a new locally restricted distance transform using the small kernel.

For the i-th iteration, let $I^{(i)}$ denote the extended binary image and let $D_I^{(i)}$ denote the local CDT of $I^{(i)}$. For the i-th iteration, we assume that the original foreground area has been extended by a margin of $i \cdot \lfloor \frac{k}{2} \rfloor$. Therefore, this offset distance is added to the current distances to

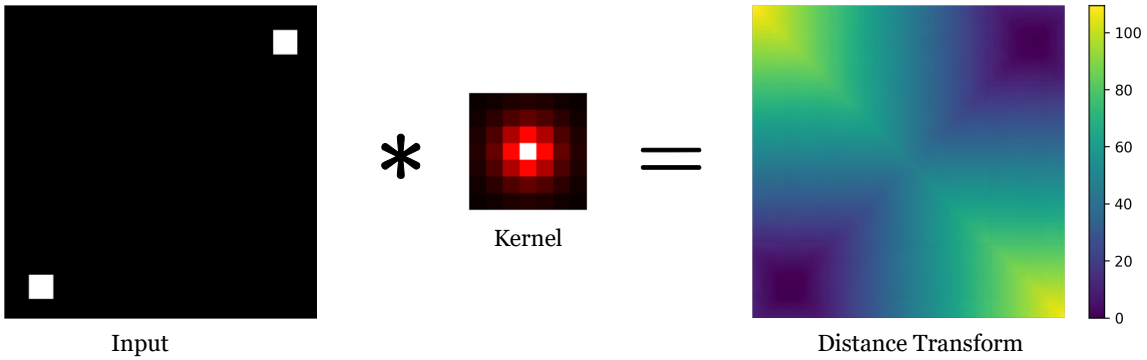


Figure 2.13: Example of the computation of a distance transform on the left image using the proposed approach.

compensate for the lower kernel size. Thus the cascaded distance transform D_I is updated by the current distances by adding $i \cdot \lfloor \frac{k}{2} \rfloor + D_i^{(i)}$ wherever $D_i(p) > 0$ holds.

This procedure can dramatically reduce the number of operations from the initial $O(w \cdot h \cdot \text{diag}^2)$ to $O(w \cdot h \cdot k \cdot \text{diag})$ if the kernel size is chosen much smaller than the image diagonal, i.e. $k << \text{diag}$. Since the maximally possible measured distance of $d(\cdot, 0)$ in 1 is restricted by the kernel size, a small kernel size additionally yields a more stable computation of the logarithmic term as the exponential does not tend to approach zero. Figure 2.13 represent the computation of the distance transform on a simple example.

2.6 Partial Conclusion

This chapter comprehensively studies active contour models and geodesic methods, showing their evolution and important role in image segmentation. Starting from the foundational work of Kass et al. [Kas88], we have explored the parametric active contour models, also called snakes. These models introduced the idea of energy-minimizing curves guided by internal and external forces to find object boundaries in images accurately. While effective in some cases, parametric models have significant limitations, such as sensitivity to initial conditions, difficulty in handling topological changes, and dependence on the curve's parameterisation.

To overcome these challenges, improvements such as the balloon model by Cohen [Coh91] introduced an inflation force, allowing the contour to expand or contract dynamically, which helps it converge better to object boundaries. The Geodesic Active Contour model by Caselles et al. [Cas97] advanced the field further by reformulating the problem using geodesic computations in a Riemannian space derived from the image. This approach reduced the dependence on parameterisation and improved the model's ability to capture complex shapes.

Then we moved to the level set method, a powerful framework introduced by Osher et al. [Osh88], which represents contours implicitly as the zero level set of a higher-dimensional function. This method naturally handles topological changes, allowing the contour to split

and merge during evolution. The level set framework provides robustness and flexibility in modelling shape evolution, as shown by the Chan-Vese model [Cha01], which segments images based on region statistics without the need for edge detection. This model demonstrated the benefits of using region-based information, improving segmentation in images with noise and weak edges.

Furthermore, we explored geodesic methods and their implementation through the Eikonal equation and the Fast Marching algorithm. These methods allow efficient computation of geodesic distances, helping to calculate minimal paths important for various image processing tasks, like segmentation and shape analysis. We also discussed the concept of distance transforms, highlighting their role in computing geodesic distances and their efficient implementation using convolution operations. We addressed computational challenges related to large kernel sizes in convolutional distance transforms and presented strategies to mitigate these issues through iterative approaches.

Throughout this chapter, we have highlighted the progression from parametric active contours to level set methods and geodesic computations, emphasising the increased flexibility, robustness, and computational efficiency these methods bring to image segmentation and shape modelling. The shift towards implicit representations and the integration of geometric principles have significantly enhanced our ability to tackle complex segmentation tasks involving intricate shapes and varying topologies.

Chapter 3

Technical Background on Deep Learning

Objectifs

Deep learning is a rapidly growing field that has significantly impacted machine learning in recent years. This chapter will introduce deep learning and explore its applications in various domains, such as image recognition, natural language processing, and predictive analytics. We will begin by discussing the fundamental components of deep learning, including supervised learning and the basic building blocks of neural networks. We will explain the strengths and weaknesses of these elements and how they contribute to the overall functionality of deep learning systems. Our focus will then shift towards the practical application of deep learning theories. Instead of delving into specific use cases, we aim to provide a comprehensive understanding of the various deep learning concepts and their implications. For readers new to these topics, this chapter will provide a foundation to understand the objectives of this thesis. We will use simple and clear language to explain complex ideas, making them accessible to a broad audience. Overall, this chapter introduces deep learning, offering insights into its key components, applications, and practical implications. By the end of this chapter, readers should have a thorough understanding of deep learning and its potential uses in various fields.

Contents

3.1 Supervised Learning

In this chapter, we will explore the fascinating world of machine learning, which involves creating algorithms that can learn from data and make predictions or decisions based on that learning. We will focus on a particular subset, supervised learning.

One of the intriguing aspects of machine learning is its ability to learn and predict things that humans cannot model or systematically solve. For example, we know how to recognise patterns in images, sounds, and texts but often need help explaining what our cognitive process does to analyse them. Similarly, when talking or listening, we do these tasks unconsciously without regard to the complex movements of our mouth and tongue.

The idea behind supervised learning is to provide the computer with as much labelled data as possible to learn to recognise patterns and make predictions on its own. This can be seen as the computer trying to extrapolate from many examples what the expected answer is (however, in practice, as we will see later, it is more interpolating in the space of examples).

Clever algorithms are designed using parametric algorithms to reach this behaviour in computers. Given a set of parameters, these algorithms leave a subset of their parameters unspecified. The learning phase involves finding the best values for these parameters. This is where the learning happens.

Most of the models we present in this thesis can be understood as parametric functions where the parameters are adjusted during the training (or learning phase) to improve the accuracy of the predictions. For example, a neural network can be seen as a big black box trained until it has reached enough confidence in predicting new unseen examples.

In the following sections, we will see that in mathematical terms, machine learning involves defining a parameter space, i.e., the space containing the possible values that a parameter can take, and measuring the accuracy of the answer to a new example to assess its capacity to generalise well.

3.1.1 Definition and Conceptual Overview

In recent years, deep learning methods have become state-of-the-art techniques for various image analysis tasks, including image classification, object detection, segmentation, and generation. However, traditional methods such as handcrafted feature extraction and machine learning algorithms are still used in specific applications. In this section, we will compare deep learning methods and conventional methods for image analysis and discuss the advantages and disadvantages of each approach.

Traditional methods for image analysis typically involve the following steps (See Figure 3.1):

1. preprocessing: enhance the image quality and remove noise;
2. feature extraction: manual design of features that capture the relevant information from the image;
3. feature selection: select the most informative features;

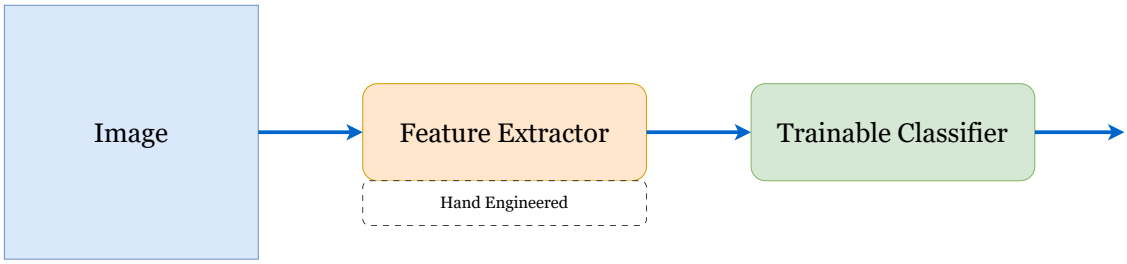


Figure 3.1: Traditional feature extraction pipeline – The figure represents the conventional approach to machine learning, where features are manually engineered before being passed to a trainable classifier for classification.

4. classification or regression.

One of the main advantages of traditional methods is their interpretability. Since the features are manually designed, it is easier to understand how the algorithm makes its decisions. Additionally, conventional methods can be more efficient regarding computational resources since they only require the computation of a small set of features. However, the performance of traditional methods is often limited by the quality of the handcrafted features, which may only capture some of the relevant information in the image.

Deep learning methods, on the other hand, automatically learn features from the raw image data using neural networks (See Figure 3.2). Several studies have compared the performance of deep learning and traditional methods for various image analysis tasks. For example, [Kri12] compared deep CNN performance with SIFT features and Fischer Vectors [Sán11] for image classification on the ImageNet dataset. The results showed that the deep CNN significantly outperformed the SVM with handcrafted features. Similarly, [Gir14] compared the performance of a deep CNN and a standard HOG-based deformable part-based model (DPM) [Gir12; Lim13; Ren13] for object detection on the PASCAL VOC dataset. The results showed that the deep CNN significantly outperformed the DPM.

Supervised learning is a critical method used in machine learning and artificial intelligence. This method provides the system with the input and the corresponding desired output data. This data is labelled, organised and classified to help the system learn effectively and serve as a foundation for processing new data.

The term supervised is akin to having a *teacher* overseeing the learning process. Here, our

Figure 3.2: Visualization of feature extraction in a deep learning model – The figure illustrates the hierarchical representation of features learned by a fully trained neural network, starting from low-level features to mid and high-level features.

student algorithm learns from the labelled data we provide. The process is considered supervised because we control the learning, knowing the expected output for each piece of input data. Figure 3.3 illustrates this for a simple classification task where we want to assign a label to an image. In this example, a cat is depicted in the picture and the supervised algorithm predicts that the object of interest is a dog. The error is corrected by updating the parameters of the learning algorithm (See Section 3.2.3) as depicted by the red arrow. Once the algorithm has been adequately trained, it can predict outputs for new, unseen input data. The performance of a supervised learning model is evaluated by how accurately it can classify new data or make accurate predictions.

In essence, supervised learning is a learning process in which we teach or train the machine using well-labelled data, meaning the data is already tagged with the correct answer. After that, the machine is provided with a new set of data, so supervised learning algorithms analyse the training data and produce an inferred function (See Figure 3.4b), which can be used to map new examples. This method allows us to train machines that can classify emails and photographs, recognise speech, and give precise predictions.

In more concrete terms, we aim to train a machine to distinguish between images of cats and dogs. In the context of supervised learning, we would feed the machine an extensive set of images, each labelled either as 'cat' or 'dog'. This dataset, known as the training set, forms the basis for the machine's learning. Each image i in the training set is a pair (x_i, y_i), where x_i is the image, and y_i is its corresponding label (cat or dog). The machine studies this training set using a machine learning algorithm and learns to map the input images x to the correct labels

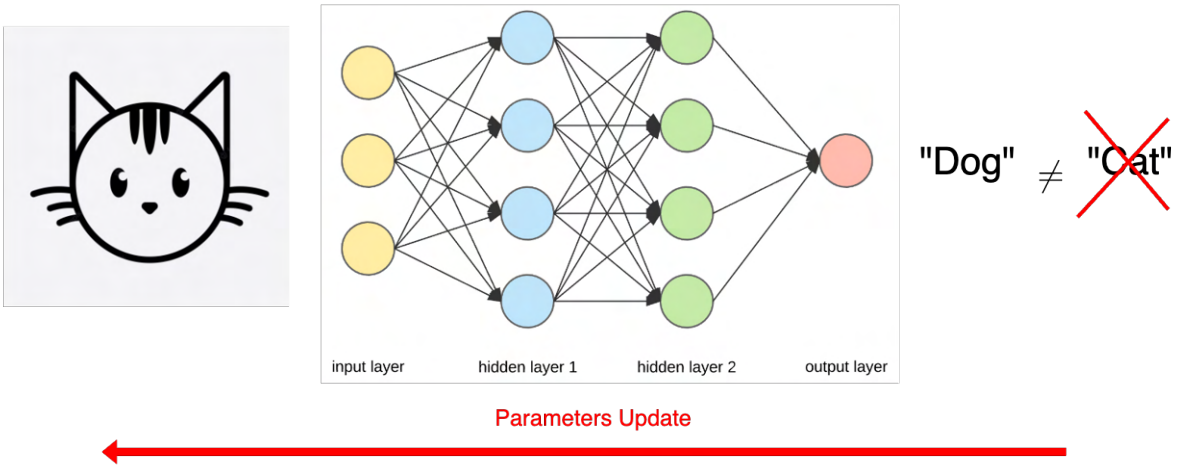
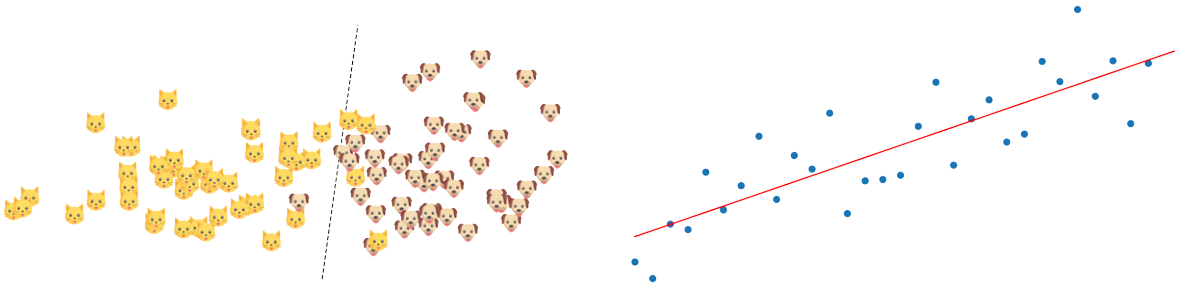


Figure 3.3: Diagram outlining the steps of supervised learning



(a) (b)

Figure 3.4: Example of two classical supervised problems: classification and regression.

y, effectively learning to identify the distinguishing features of cats and dogs (See Figure 3.4a). Figure 3.4a represents the separation created by the machine learning algorithm after training, in this case, a linear model.

The underlying mathematical formality of this learning process can be expressed as learning a function f that best maps input variables $\mathbf{x}$ to an output variable $\mathbf{y}$, given by: $\mathbf{y} = f(\mathbf{x})$. The algorithm learns the function based on the input-output pairs from the training data. Once the function is learned adequately, it can predict the output for new, unseen input data.

Within the broad landscape of supervised learning, problems are typically divided into Classification and Regression. Classification involves predicting discrete labels, such as "spam" or "not spam" for emails or "cat" or "dog" for our animal images. Mathematically, a classification model learns a function that maps the input to discrete categories.

On the other hand, regression problems involve predicting a continuous outcome variable. For instance, they could predict stock market prices or determine an individual's age based on specific features. In mathematical terms, a regression model learns a function that maps the input to continuous values. While the techniques used to solve classification and regression problems can often be very similar, the type of problem fundamentally changes how we evaluate model performance.

(a) (b)

Figure 3.5: Example of unsupervised problems: reinforcement learning and k-means.

While supervised learning serves as a strong tool, it's crucial to note that it's only one of several learning strategies employed in machine learning. Unsupervised learning, for example, involves training models using data that isn't classified or labelled. The algorithm can discern and act on underlying patterns in the data without external knowledge. Figure 3.5b shows three clusters identified by the algorithm, each cluster corresponding to a distinct grouping of data points based on their similarity in feature space, likely representing different species in the Iris dataset [Fis36]. Another paradigm, self-supervised learning, learns representations from the data itself by training itself to understand one part of the input from another part of the input. In contrast, reinforcement learning involves training an agent to make specific decisions based on reward and penalty; desirable actions are rewarded, and undesirable ones are penalised. Figure 3.5a) illustrate how the learning agent, here Mario, interacts with his environment to improve over time. Through actions, observations, and rewards, the agent learns an optimal policy by exploring the environment, adjusting its behaviour based on feedback, and refining its decision-making process to maximize its long-term rewards in following trials. These diverse techniques highlight the rich and varied nature of machine learning, each offering unique ways to extract patterns and insights from data.

When we look at the big picture of artificial intelligence (AI), supervised learning is a critical way that machines can learn to make predictions, make decisions, and understand patterns. It's a building block for creating complex AI systems that can act like humans. The influence of supervised learning is everywhere already in our digital lives. For instance, supervised learning algorithms are at play when an online platform recommends a movie based on your previous viewing habits. Similarly, when your voice assistant accurately processes your verbal requests or a self-driving vehicle correctly identifies traffic signs, these are examples of supervised learning.

3.1.2 Supervised Learning Tasks

The *input space* or *feature space* is represented by a set $\mathcal{M}$. An element of $\mathcal{M}$ is denoted as $\mathbf{x}$, which refers to the input to our learning system. In the case of a real-world problem such as

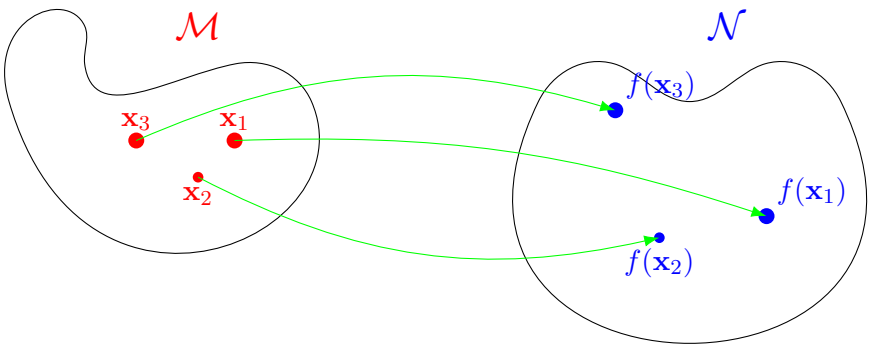


Figure 3.6: A neural network transforms the initial representation in a space $\mathcal{M}$, the feature space, to a simpler representation as an element of $\mathcal{N}$, the label space.

object recognition, $\mathbf{x}$ could be an image. For simplification and the following computations, we assume that our inputs $\mathbf{x}$ are numerical or have been transformed to numerical representation as this is the form in which they are stored and processed in a computer. Therefore, $\mathcal{M}$ is usually a subset of $\mathbb{R}^d$.

The *output space* or *label space* is represented by a set $\mathcal{N}$. An element of $\mathcal{N}$ is denoted as $\mathbf{y}$ and refers to the output or the label associated with an input. In the case of object recognition, y could be a label representing the object in the image, such as a car or plane. For example, in classification problems where we need to classify inputs into distinct categories, $\mathcal{N}$ is a discrete set, whereas in regression problems where we predict a continuous value, $\mathcal{N}$ is a subset of the real numbers.

A training set is a set of pairs $\{(x_i, y_i)_{i\in[1:n]}\}$ where the x_i are a sequence of inputs and y_i are the corresponding sequence of labels. The goal of supervised learning is to use the training set to learn a function $f : \mathcal{M} \to \mathcal{N}$ that can accurately predict the label $\mathbf{y}$ for a new input $\mathbf{x}$. The function f is learned through an algorithm that adjusts its parameters to minimise a loss function $L(\mathbf{y}, f(\mathbf{x}))$, which measures the discrepancy between the true labels y and the predictions $f(\mathbf{x})$. The function f learns to transform the initial representation into a simpler representation to classify the elements (See Figure 3.6).

Note that the function f and the loss function L depend on the type of problem and the model chosen for the task. For example, in the case of linear regression, f is a linear function of the input features, and L could be the mean squared error between the actual labels and the predictions.

The goal of training a machine learning model is to find the best parameters θ for a function $f_\theta : \mathcal{M} \to \mathcal{N}$ that maps inputs to outputs or predictions. This function, f_θ, represents our machine learning model. Depending on the context and the type of problem, this function can take various forms, such as a linear function in linear regression or a more complex function in deep learning models. The set of all possible values of the parameters θ forms the parameter space, denoted as Θ. Typically, the parameters are numerical values, which implies that Θ is a subset of $\mathbb{R}^d$.

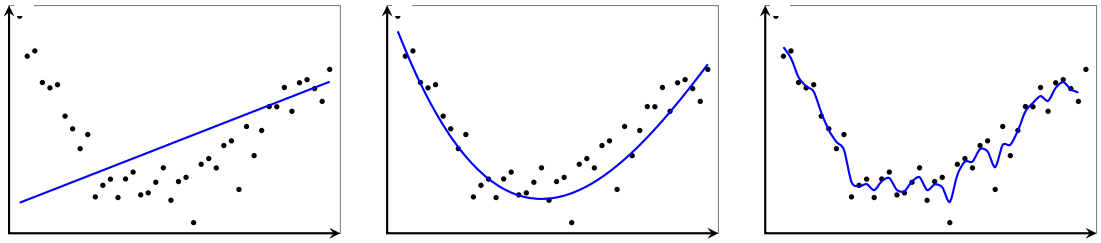


Figure 3.7: Three examples of regression problems showing the respectively from left to right under fitting, balance between bias and variance, and overfitting

In an ideal scenario, where our model has learned the exact relationships between the input features and the output, the predictions made by the model for all inputs in the training set would match the actual labels, i.e., for all $\mathbf{x}_i$ in X, we would have $f_\theta(\mathbf{x}_i) = \mathbf{y}_i$.

However, real-world data is usually noisy and may contain inaccuracies. If we fit our model to such a noisy training set perfectly, it could learn to reproduce the noise and inaccuracies in the data. This would lead to a model that performs well on the training data but fails to generalise to unseen data, a problem known as overfitting. Thus, machine learning aims to find a balance between fitting the training data and preserving the ability to generalise to new, unseen data. This often involves allowing some level of error in the predictions of the training data. The exact balance is determined by the complexity of the model, the amount and quality of the training data, and the method used to train the model (See Figure 3.7).

Training datasets can contain inaccuracies in the form of noise, bias, corruption, or incompleteness, each presenting unique challenges.

Noise, or random errors in the data, can hide the underlying patterns a model needs to learn, leading to a model needing help generalising to new data. For example, if training a model to identify handwritten digits from images, the images collected could vary in lighting conditions, introducing random brightness variations in the data. The model might need help to learn the correct patterns, as the brightness variations could overshadow the actual patterns of the digits.

Systematic errors, or bias, can skew the model's understanding of these patterns, resulting in similar issues with generalisation. When building a facial recognition model, if all the faces in the training data are of people from one specific age group, the model will perform poorly on an other age group. The model might be great at recognising faces within that age group but fails when trying with faces from other age groups. This isn't very objective, as the model has learned a pattern specific to one age group because the training data was not representative of the entire population.

Corruption in the data due to errors in collection or processing results in false or misleading data that can disrupt the model's learning process and obstruct the interpretability of its predictions. If we train an image classifier and some images get distorted or flipped during preprocessing or the labels get swapped accidentally, the model might learn incorrect patterns. For instance, if a dog image is labelled a cat, the model could start associating dog features with cats.

Similarly, incompleteness, where data is missing or records are incomplete, can prevent the

model from thoroughly learning the underlying patterns, leading to errors when predicting new data. While only sometimes affecting performance on the training data, these inaccuracies can cause generalisation errors where the model performs well on training data but poorly on new, unseen data. This typically happens because the model learns the inaccuracies present in the training data rather than the underlying patterns.

Regression In the context of linear regression, we are given a set of pairs $\{(x_i, y_i)_{i\in[1:n]}\} \in (\mathbb{R}^{l\times 1})^n$, obtained from, for example, computer simulations or experimental data. Regression analysis aims to discover a relationship between these vectors x and y. We define:

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad x = \begin{pmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^n \end{pmatrix} i = 1, \cdots, l \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

where ε is the matrix of biases.

We then denote $\psi(x)$ as the vector that applies ψ_i to each component x_i. The function space, denoted by F, is composed of linear combinations of basis functions ψ_i and we write:

$$\psi(X) = \begin{pmatrix} \psi_1(x_1)|\psi_2(x_2)|\cdots|\psi_l(x_l) \end{pmatrix}$$

The least squares method provides an estimator for the output vector $\hat{y}_z = f(z, w), \forall z \in \mathbb{R}^n$, where z is a new observation, intended to be as close as possible to y, where w are the parameters to optimise. In this section, we denote f as the functions from the chosen function family for our linear model, x as the input vector of assumed size n, and w as the parameter vector of size l. We aim to learn an approximation for a function from $\mathbb{R}^n \to \mathbb{R}$. :

$$F_z = \{f; f(z, w) = \sum_{i=1}^{l} \psi_i(z_i) w_i\} \tag{3.1}$$

This set can be visualised on a plane or as a scatter plot. We can rewrite the expression for function f by assuming that the functions ψ_i are independent of the model parameters.

$$Y = \psi(X)^T w + \varepsilon$$

with $w \in \mathbb{R}^l$, $\varepsilon \in \mathbb{R}^n$, and $X \in M_{n,l}$ with $Rank(X) = l$.

The functions ψ could be, for example, polynomials. The system can be written as:

$$\begin{cases} y_1 = w_1\psi_1(x_1) + w_2\psi_2(x_1) + \ldots + w_l\psi_l(x_1) + \varepsilon_1 \\ y_2 = w_1\psi_1(x_2) + w_2\psi_2(x_2) + \ldots + w_l\psi_l(x_2) + \varepsilon_2 \\ \vdots \\ y_n = w_1\psi_1(x_n) + w_2\psi_2(x_n) + \ldots + w_l\psi_l(x_n) + \varepsilon_n \end{cases} \tag{3.2}$$

To express the empirical risk in terms of matrices, the model matrix X, where each row

corresponds to an example from the training base, and each column represents a parameter of the model expressed in the base of secondary variables (the ψ_i). Thus, we have:

$$X_{ij} = \psi_j(x_i) \quad \Rightarrow \quad R_{\text{emp}}(f) = \frac{1}{n}\sum_{i=1}^{n}(f(x_i, w) - y_i)^2 \tag{3.3}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(\psi(x_i)^T w - y_i)^2 = \frac{1}{n}(Xw - y)^T(Xw - y) \tag{3.4}$$

Our goal is to minimise the error:

$$\min ||Xw - y||^2 \tag{3.5}$$

The derivative of this function with respect to w is:

$$\nabla_w \|Xw - y\| = 2X^T(Xw - y). \tag{3.6}$$

The minimum is reached when $X^T Xw = X^T y$. Applied to empirical risk, we get:

$$\frac{\partial R_{\text{emp}}(f)}{\partial w} = \frac{2}{n}(X^T Xw - X^T y) \tag{3.7}$$

The minimum is reached at the value we denote as w_s: $w_s = (X^T X)^{-1} X^T y$. A simple geometric interpretation is possible considering the observation space, which the Gram-Schmidt process can orthonormalise. The solution to our least squares problem appears as the orthogonal projection matrix onto the solution subspace (the basis being the columns of the matrix X).

Given $X \in M_{n,l}$ with $n > l$ and $y \in \mathbb{R}^n$, we say that the vector $w \in \mathbb{R}^l$ minimizes $||Xw - y||^2$ if and only if $X^T Xw = X^T y$. These are called normal equations; this system admits at least one solution. Finally, if $X^T X$ is regular (i.e., rank$(X) = l$), then the solution is unique.

While not immediately related to linear regression, deep learning can be seen as a generalisation. In a deep learning model, instead of using simple basis functions ψ_i and a linear combination of them, we use artificial neural networks, which can represent highly complex functions. Deep learning generalises linear regression to non-linear and high-dimensional contexts.

Classification Classification represents a distinctive approach to predictive modelling, which inherently differs from regression. Instead of predicting continuous numeric values, the goal of classification is to predict labels that are categorical and unordered. The labels, or 'classes', stem from a predetermined set which does not entail a natural hierarchical structure. A typical example is binary classification, deciding whether a human face is present in an image. Nevertheless, classification is open to more than a binary context and can handle problems with multiple classes reaching hundreds or thousands.

The critical distinction between regression and classification is that the output of the predictive function f_θ is discrete in classification. Directly optimising for this discrete output would

lead to a combinatorial problem. To avoid this, most approaches opt for a relaxation of the output constraint during the optimisation process, and later, the obtained continuous values are converted back into discrete class labels. For instance, consider a binary classification scenario where the classes are denoted as 0 and 1. The outputs of f_θ can be taken as probabilities in the continuous range [0,1].

1. Thresholding is a common strategy to convert these continuous outputs into discrete class labels. For example, any output value $p < 0.5$ is mapped to class 0, while any $p \geq 0.5$ is mapped to class 1.

2. This approach lends itself to an interpretation of $f_\theta(x_i)$ as the probability of x_i belonging to class 1.

This probabilistic framework can be used to formulate the loss function for the classification model. Let $f_\theta(x_i) = p_i$ for each i, and treat the vector p as a distribution over the space of classes M. The ground truth labels y can similarly be interpreted as a distribution over M. The loss function can then be defined as a divergence measure between the two distributions p and y. A standard choice for this divergence measure is the cross entropy loss:

$$CE(y, p) = -\sum_{i=1}^{n} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \tag{3.8}$$

When y and p fall in the range [0,1], the summand in the cross entropy loss is non-positive and is zero when $p_i = y_i$. Given that p depends on the model parameters θ, the cross entropy provides a suitable choice for the loss function $L(\theta)$ to be minimised during the learning process.

The described classification setup is highly relevant in the context of deep learning. Deep learning models for classification often involve an output layer with a softmax activation function, which produces output probabilities summing to one. The soft(arg)max (we want the most probable position and not the probability) allows us to transform a vector of k components into a distribution over k classes. The function is defined as :

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{i=0}^{k} e^{z_i}} \forall j \in \{1, \cdots, k\}. \tag{3.9}$$

These models are trained by minimising the cross entropy loss between the predicted class probabilities and the actual class labels. This makes deep learning a powerful tool for tackling classification problems, providing state-of-the-art results in many domains, including image classification, text categorisation, and more.

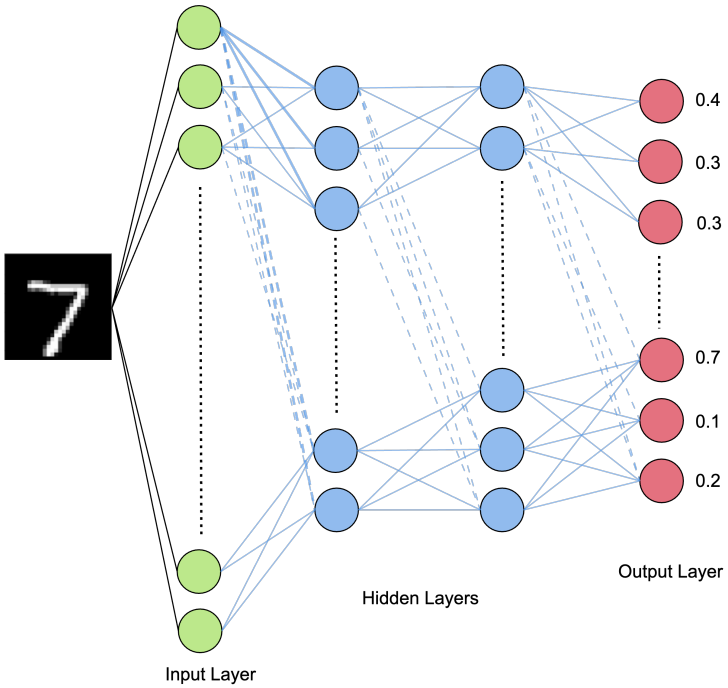


Figure 3.8: Neural Network given an example from the MNIST dataset. The Neural Network has 2 hidden layers.

3.2 Neural Networks

3.2.1 Introduction to Neural Networks - An overview of neural networks and their origins

Here are some key terms that will appear multiple times throughout this thesis:

- **Batch:** A batch refers to a small group of data samples processed together by a machine learning model before updating its parameters. Instead of feeding all the data at once, batches make the training more manageable and efficient.
- **Epochs:** An epoch is one complete pass through the entire training dataset. During each epoch, the model processes all the training data once, typically in smaller batches, and updates its parameters to improve performance.
- **Learning Rate:** The learning rate is a parameter that controls how much the model's parameters are adjusted during training. It determines the step size for each update. A higher learning rate means larger updates, while a lower learning rate results in smaller, more gradual updates.

Neural Network (NN) is a machine learning algorithm inspired by the human brain's struc-

ture and function. We want to mimic the connections and the information-passing mechanism in the brain. The idea behind neural networks dates back to the 1940s and 1950s from McCulloch et al. [McC43] when researchers first began to explore the possibility of building artificial systems that could perform tasks like image and speech recognition, which are easy for humans but difficult for computers.

The basic block of a *Neural Network* (NN) is a neuron modelled after the human brain's neurons. Each neuron takes in one or more inputs (an electric impulsion in the brain), applies a mathematical function, and produces an output. The output of one neuron can be used as the input to another neuron, allowing the network to perform increasingly complex computations.

Usually, we organise *Neural Network* (NN) into layers, each containing a set of neurons. The first layer, the input layer, receives the raw data the network will use to make predictions. The last layer is the output layer, which produces the final network prediction. Between the input and output layers, there can be as many layers as one wants; we call them hidden layers, and they perform intermediate computations and help the network learn complex patterns in the data.

To illustrate the notion and how a NN works, we can consider a simple example. Suppose we want to create a model that can learn to recognise handwritten digits [LeC89]. If each image is of size 28×28, we can make a first layer containing 784 neurons, one for each pixel. The output layer will contain 10 neurons, one for each possible digit (0 through 9). We can have one or more hidden layers between the input and output layers with a variable number of neurons (See Figure 3.8). During training, we present as many examples of handwritten digits as possible to the NN and their corresponding labels. (i.e. the correct digit for each image). The network adjusts the strength of the connections between neurons (called weights) to minimise the difference between its predicted outputs and the actual labels.

3.2.2 Deep Neural Networks

This section describes the formulation of a *Neural Network* (NN). We define the input and output of the NN as layers that we denote $x^{(0)}$ and $x^{(L)}$, $L+1$ is the number of layers in the neural network. We define the hidden layers as $x^{(l)}$ with $l \in \{1, \cdots, L-1\}$. We subscript the n neurons in the fist layer $x_n^{(l)}$ and m neurons in the second one $x_m^{(1)}$. Between each layer we find weights $w^{(l)}$ and bias $b^{(l)}$ and we write the connexion between two neurons $i = \{1, \cdots, m\}$ and $j = \{1, \cdots, n\}$ as $w_{i,j}^{(l)}$. Between each layer, we find an activation function σ. Figure 3.9 summarises the computation between each layer.

3.2.3 Backpropagation: The Mathematical Backbone of Deep Learning Optimization

There are multiple methods to compute derivatives numerically. Among them are finite difference methods. It performs derivatives using the definition in terms of limits, substituting a small increment h to approximate the derivative.

$$\frac{\partial}{\partial x_i} f(x_1, \cdots, x_N) \approx \frac{f(x_1, \cdots, x_i + h, \cdots, x_N) - f(x_1, \cdots, x_i, \cdots, x_N)}{h}. \tag{3.10}$$

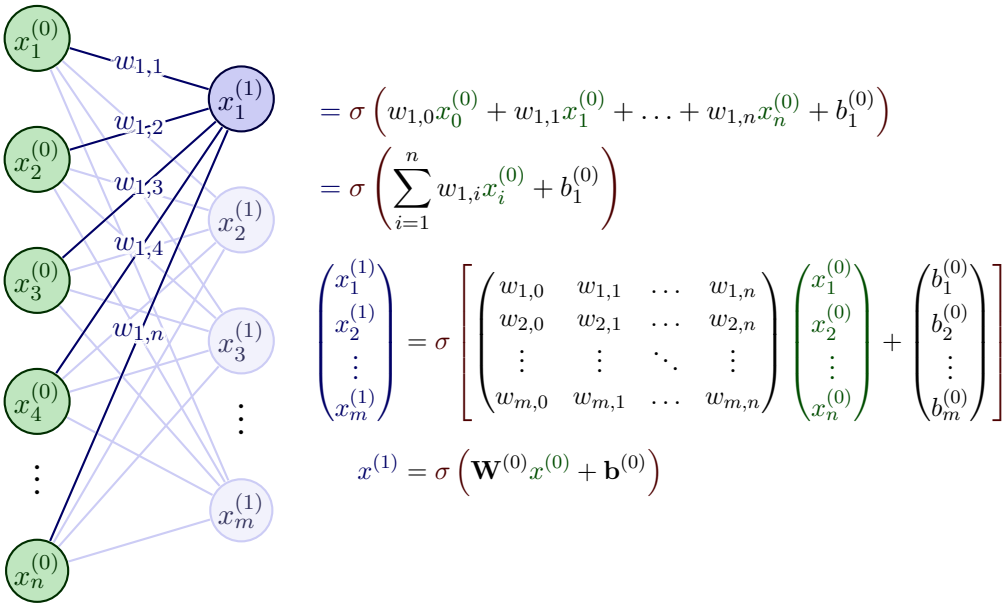


Figure 3.9: Example of the connections between two layers of a fully connected neural network.

We usually use these methods to validate gradient computations. Despite that, they are straightforward to use as they only require function evaluation but they are impractical to train neural networks due to the high computational costs, requiring a separate forward pass for each partial derivative, and numerical instability from subtracting nearly equal values and dividing by a small h.

Backpropagation is one of the foundational algorithms in deep learning. It enables the efficient computation of gradients, a prerequisite for optimising a neural network's parameters. The essence of backpropagation lies in an elegant application of the chain rule from calculus. It is a technique used to compute the partial derivatives of a loss function with respect to the parameters of a neural network. These derivatives are then employed in gradient descent, similar to their use in linear and logistic regression.

3.2.3.1 A simple univariate example

Before presenting backpropagation in the case of Neural Networks, we start by looking at a simple example. We consider a single input example $(x, \hat{x})$ and the predictions are made using a linear operation with a sigmoid activation function to introduce non-linearity.

$$z = wx + b \tag{3.11}$$
$$y = \sigma(z) \tag{3.12}$$
$$\mathcal{L} = \frac{1}{2}(y - \hat{x})^2 \tag{3.13}$$

We want to compute the partial derivatives of the cost function in terms of w and b. To do so, we compute the derivatives and the chain rule multiple times.

Definition 5 (Chain rule (2 variables case))**.** Given a function f of 2 variables, we want to compute $\frac{d}{dt}f(x(t), y(t))$. The chain rule gives:

$$\frac{d}{dt}f(x(t), y(t)) = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} \tag{3.14}$$

In the case of our simple example in the univariate case, we have:

$$\mathcal{L} = \frac{1}{2}(\sigma(wx+b) - \hat{x})^2 \tag{3.15}$$

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial}{\partial w}\left[\frac{1}{2}(\sigma(wx+b) - \hat{x})^2\right] \tag{3.16}$$

$$= (\sigma(wx+b) - \hat{x})\frac{\partial}{\partial w}\left[(\sigma(wx+b) - \hat{x})\right] \tag{3.17}$$

$$= (\sigma(wx+b) - \hat{x})\sigma'(wx+b)\frac{\partial}{\partial w}(wx+b) \tag{3.18}$$

$$= (\sigma(wx+b) - \hat{x})\sigma'(wx+b)x \tag{3.19}$$

$$\tag{3.20}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial}{\partial b}\left[\frac{1}{2}(\sigma(wx+b) - \hat{x})^2\right] \tag{3.21}$$

$$= (\sigma(wx+b) - \hat{x})\frac{\partial}{\partial b}\left[(\sigma(wx+b) - \hat{x})\right] \tag{3.22}$$

$$= (\sigma(wx+b) - \hat{x})\sigma'(wx+b)\frac{\partial}{\partial b}(wx+b) \tag{3.23}$$

$$= (\sigma(wx+b) - \hat{x})\sigma'(wx+b) \tag{3.24}$$

$$\tag{3.25}$$

This provides a clear example of the several drawbacks inherent in this method. Firstly, the calculations are exceedingly cumbersome. Throughout this derivation, numerous terms had to be transcribed from one line to the next, making it easy to omit something inadvertently. Although the calculations are manageable in this simplified example, they become overwhelmingly complex for realistic neural networks. Secondly, the calculations entail a considerable amount of redundant work. For example, the initial three steps in the two derivations above are nearly identical. Thirdly, the final expressions contain numerous repeated terms, resulting in substantial redundancy if implemented directly. For instance, $wx + b$ is computed a total of four times between $\frac{\partial \mathcal{L}}{\partial w}$ and $\frac{\partial \mathcal{L}}{\partial b}$. The larger expression $(\sigma(wx+b) - t)\sigma'(wx+b)$ is computed twice. Recognising these redundancies might allow for a more efficient implementation by factoring out the repeated expressions.

The fundamental idea behind backpropagation is to leverage these repeated computations wherever possible. When executed correctly, backpropagation calculations are notably clean and modular.

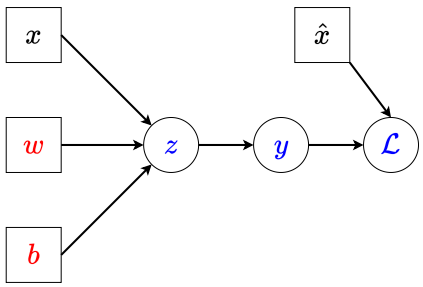


Figure 3.10: Computational graph for a linear regression.

3.2.3.2 The Computation Graph

To compute the derivatives computationally, we represent all the variables and operations as a graph. For the example in section 3.2.3.1, we construct the graph presented in Figure 3.10.

As we can see, we have a directed acyclic graph.

The nodes in the graph represent all the computed values, with edges indicating the dependencies between these values.

The objective of backpropagation is to compute the derivatives with respect to w and b. This is achieved by repeatedly applying the Chain Rule. To calculate a derivative using Equation 3.14, the derivatives of its child nodes in the computation graph must be known first. Consequently, we must begin from the final result of the computation (in this case, $\mathcal{L}$) and work backwards through the graph. This backward traversal is why backpropagation and reverse mode autodifferentiation are named as such.

We begin with the formal definition of the algorithm. Let $v_1, \ldots, v_N$ represent all the nodes in the computation graph arranged in a topological order. A topological ordering ensures that parent nodes precede their child nodes. We aim to compute the derivatives of all $\frac{\partial \mathcal{L}}{\partial v_i}$, though we might only be interested in a subset of these values. Initially, all values are computed in a forward pass, followed by the computation of derivatives in a backward pass. Specifically, v_N denotes the final result of the computation, and it is the quantity for which we seek to compute the derivatives. By convention, we set $\frac{\partial \mathcal{L}}{\partial v_N} = 1$. The algorithm proceeds as follows:

1. For $i = 1, \ldots, N$:
   - Compute v_i as a function of $Pa(v_i)$
2. Set $\frac{\partial \mathcal{L}}{\partial v_N} = 1$
3. For $i = N - 1, \ldots, 1$:
   - $\frac{\partial \mathcal{L}}{\partial v_i} = \sum_{j \in Ch(v_i)} \frac{\partial \mathcal{L}}{\partial v_j} \frac{\partial v_j}{\partial v_i}$

Here, $Pa(v_i)$ denotes the parents of v_i, and $Ch(v_i)$ denotes the children of v_i. This algorithm ensures the efficient computation of derivatives by leveraging the structure of the computation graph.

Let's use the graph to compute the algorithm for the linear regression example in section 3.2.3.1.

$$\frac{\partial \mathcal{L}}{\partial \mathcal{L}} = 1 \tag{3.26}$$

$$\frac{\partial \mathcal{L}}{\partial y} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}y} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}}(y - t). \tag{3.27}$$

$$\frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}}{\partial y} \frac{\mathrm{d}y}{\mathrm{d}z} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}}(y - t)\sigma'(z). \tag{3.28}$$

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial z} \frac{\mathrm{d}z}{\mathrm{d}w} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}}(y - t)\sigma'(z)x. \tag{3.29}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial z} \frac{\mathrm{d}z}{\mathrm{d}b} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}}(y - t)\sigma'(z)b. \tag{3.30}$$

$$\tag{3.31}$$

We remove most of the computation using the proposed algorithm, and each previous computation is reusable.

3.2.3.3 A simple Neural Network

We can now use the same algorithm for a small neural network with two inputs, one output, and two hidden layers (see Figure 3.11). We write the operations performed by the neural networks as follows:

The operation performed by the neural network are:

$$z_i = \sum_j w_{ij}^{(1)} x_j + b_i^{(1)} \tag{3.32}$$

$$h_i = \sigma(z_i) \tag{3.33}$$

$$y_k = \sum_i w_{kj}^{(2)} h_i + b_k^{(2)} \tag{3.34}$$

$$\mathcal{L} = \frac{1}{2} \sum_k (y_k - \hat{x}_k)^2. \tag{3.35}$$

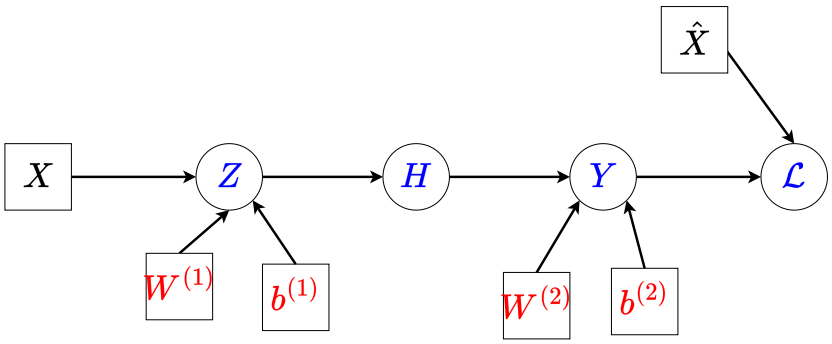


Figure 3.11: Computational graph for two inputs, one output and two hidden layers.

We compute the derivatives with regards to the different variables in the neural network:

$$\frac{\partial \mathcal{L}}{\partial \mathcal{L}} = 1 \tag{3.36}$$

$$\frac{\partial \mathcal{L}}{\partial y_k} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}y_k} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}}(y_k - \hat{x}_k) \tag{3.37}$$

$$\frac{\partial \mathcal{L}}{\partial w_{ki}^{(2)}} = \frac{\partial \mathcal{L}}{\partial y_k} \frac{\mathrm{d}y_k}{w_{ki}^{(2)}} = \frac{\partial \mathcal{L}}{\partial y_k} h_i \tag{3.38}$$

$$\frac{\partial \mathcal{L}}{\partial b_k^{(2)}} = \frac{\partial \mathcal{L}}{\partial y_k} \tag{3.39}$$

$$\frac{\partial \mathcal{L}}{\partial h_i} = \sum_k \frac{\partial \mathcal{L}}{\partial y_k} w_{ki}^{(2)} \tag{3.40}$$

$$\frac{\partial \mathcal{L}}{\partial z_i} = \frac{\partial \mathcal{L}}{\partial h_i} \sigma'(z_i) \tag{3.41}$$

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(1)}} = \frac{\partial \mathcal{L}}{\partial z_i} x_j \tag{3.42}$$

$$\frac{\partial \mathcal{L}}{\partial b_i^{(1)}} = \frac{\partial \mathcal{L}}{\partial z_i}. \tag{3.43}$$

Now in the matrix form we have:

$$\mathbf{z} = W^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \tag{3.44}$$
$$\mathbf{h} = \sigma(\mathbf{z}) \tag{3.45}$$
$$\mathbf{y} = W^{(2)}\mathbf{h} + \mathbf{b}^{(2)} \tag{3.46}$$
$$\mathcal{L} = \frac{1}{2}(\mathbf{y} - \hat{\mathbf{x}})^2 \tag{3.47}$$

$$\frac{\partial \mathcal{L}}{\partial \mathcal{L}} = 1 \tag{3.48}$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{y}} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}}(\mathbf{y} - \hat{\mathbf{x}}) \tag{3.49}$$
$$\frac{\partial \mathcal{L}}{\partial W^{(2)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{y}}\mathbf{h}^T \tag{3.50}$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(2)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{y}} \tag{3.51}$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{h}} = {W^{(2)}}^T \frac{\partial \mathcal{L}}{\partial \mathbf{y}} \tag{3.52}$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}} = \frac{\partial \mathcal{L}}{\partial \mathbf{h}} \circ \sigma'(\mathbf{z}) \tag{3.53}$$
$$\frac{\partial \mathcal{L}}{\partial W^{(1)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}}\mathbf{x}^T \tag{3.54}$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(1)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}} \tag{3.55}$$

In this section, we demonstrated how to apply backpropagation to a simple neural network with two inputs, two hidden layers, and one output (refer to Figure 3.11). We outlined the forward pass by specifying the operations performed at each layer, including the calculation of weighted sums, the application of the activation function σ, and the computation of the loss $\mathcal{L}$.

We then derived the gradients of the loss function with respect to the network's parameters/weights and biases, both in scalar and matrix forms. By computing these derivatives, we established how the gradients propagate backwards through the network, enabling the adjustment of parameters using optimisation algorithms like gradient descent.

This detailed walkthrough illustrates the fundamental mechanisms underlying neural network training. Understanding these calculations is crucial for more complex architectures and optimisation techniques in Deep Learning. The methodologies applied here form the backbone of neural network learning, highlighting how iterative updates based on gradient information improve performance and minimise the loss over time.

3.2.4 Varieties of Neural Networks: A Brief Overview

In recent years, the field of *Deep Learning* (DL) has revolutionised various domains such as *Computer Vision* (CV), *Natural Language Processing* (NLP) and speech recognition. Deep Learning algorithms automatically learn hierarchical feature representations from large-scale data,

enabling them to achieve state-of-the-art performance in a wide range of tasks. Several types of deep learning architectures, each with strengths and weaknesses, have been developed to tackle different problems. We will focus on specific architectures recurrently found in image analysis, *Convolutional Neural Networks* (CNN), and Transformers.

Convolutional Neural Networks (CNN) is one of the most popular deep learning architectures for computer vision tasks [Fuk80; Wai13; LeC89]. CNN exploit the spatial correlations found in images using convolutional layers. The convolutional layer applies a set of learnable filters to the input image to extract local features. The filters learn to recognise specific patterns, such as edges or corners. They are shared across the entire image, significantly reducing the number of learnable parameters. CNN also typically include pooling layers which downsample the feature maps to reduce the computational complexity and improve the model's invariance to local translations.

Transformers are a more recent deep learning architecture that has achieved state-of-the-art performance in various *Natural Language Processing* (NLP) tasks [Vas17]. They have been adapted for computer vision tasks by Dosovitskiy et al. [Dos20], known as *Vision Transformers* (ViT). Transformers use a self-attention mechanism to model relationships between input sequences/image patches, allowing them to capture long-term dependencies.

3.2.4.1 *Convolutional Neural Networks* (CNN)

We will present the convolutional networks for 2D data. Convolutional Neural Networks are based on the *convolution* operation. The network inputs an image X and outputs an image Z. At each pixel (i, j), the output $z_{i,j}$ is a weighted sum of nearby pixels of x. We apply the same weights at every position. These weight patches are called convolutional kernels or filters. A kernel size defines the size of these patches or regions. For, say, a kernel size of 3, we have :

$$z_{i,j} = \sum_{k,l=1}^{3} w_{k,l} x_{i+k-2,j+l-2}. \tag{3.56}$$

Therefore, the kernel can be written as a matrix:

$$\begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} \end{pmatrix} \tag{3.57}$$

A convolutional layer is then defined as the combined operation of computing the convolution of the input with the filter and adding a bias β before passing the result through an activation function σ. Figure 3.12 summarise the computation of the convolution operation. We can write:

$$h_{i,j} = \sigma \left(\beta + \sum_{k,l}^{3} w_{k,l} x_{i+k-2,j+l-2} \right) \tag{3.58}$$

We find three main applications of CNN in computer vision: image classification, where the goal is to assign the image to one of a set of categories. Semantic segmentation, where the

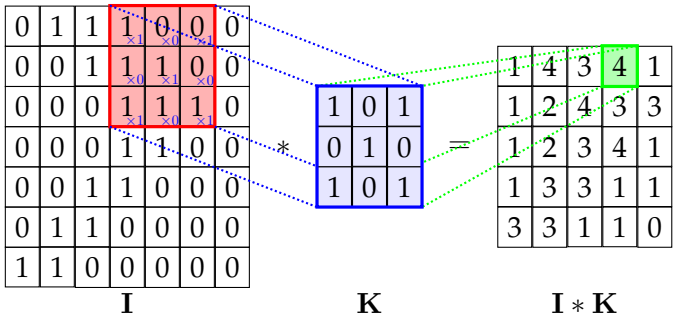


Figure 3.12: The diagram illustrates an example of a convolution operation between an input matrix I and a kernel K. The kernel slides over the input matrix, performing element-wise multiplication with the overlapping values, and the results are summed to produce the corresponding value in the output matrix $I * K$. The red-highlighted section in the input matrix represents the region currently convolved with the kernel, and the green-highlighted section shows the resulting value in the output matrix after applying the convolution at that position.

goal is to assign each image pixel to a label corresponding to an object. Object detection, where the goal is to find a bounding box around the objects of interest.

Image classification In the early days, image classification was made from hand-engineered features such as Histogram of Oriented Gradients (HOG) [Dal05] and Scale-Invariant Feature Transform (SIFT) [Low99]. However, these features have limited representational power and require significant domain expertise.

The first success of CNN, which brought them back into the spotlight, was for image classification with AlexNet [Kri12] introduced in 2012 by Krizhevsky et al. AlexNet significantly won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Ber10]. They won by a significant margin, beating the second with a top-5 error rate of $15,3\%$ while the second was at $26,2\%$. AlexNet consisted of eight layers, including five convolutional layers and three fully connected layers. It also introduced the ReLU (Rectified Linear Unit) activation function, which significantly speeds up training time compared to traditional sigmoid or tanh activation functions.

Since the introduction of AlexNet, many significant developments have been in the architecture of CNN for image classification. Some notable examples:

- VGGNet [Sim14]: Introduced in 2014 by Simonyan et al., VGGNet increased the depth of the network to 16 or 19 layers. They also made use of smaller convolutional kernels (3×3). The model showed that increasing the depth of the network improves the performance.
- GoogLeNet [Sze15]: This was also introduced in 2014 by Szegedy et al. [Sze15]. The model GoogLeNet introduced the Inception module, which allows multiple parallel convolutional filters with different sizes to be used within the same layer. This reduces the number of parameters and increases the network's representation power.
- ResNet [He16]: Introduced in 2016 by He et al., ResNet solves the problem of vanishing gradients in deep networks by introducing residual connections. It allows direct gradient

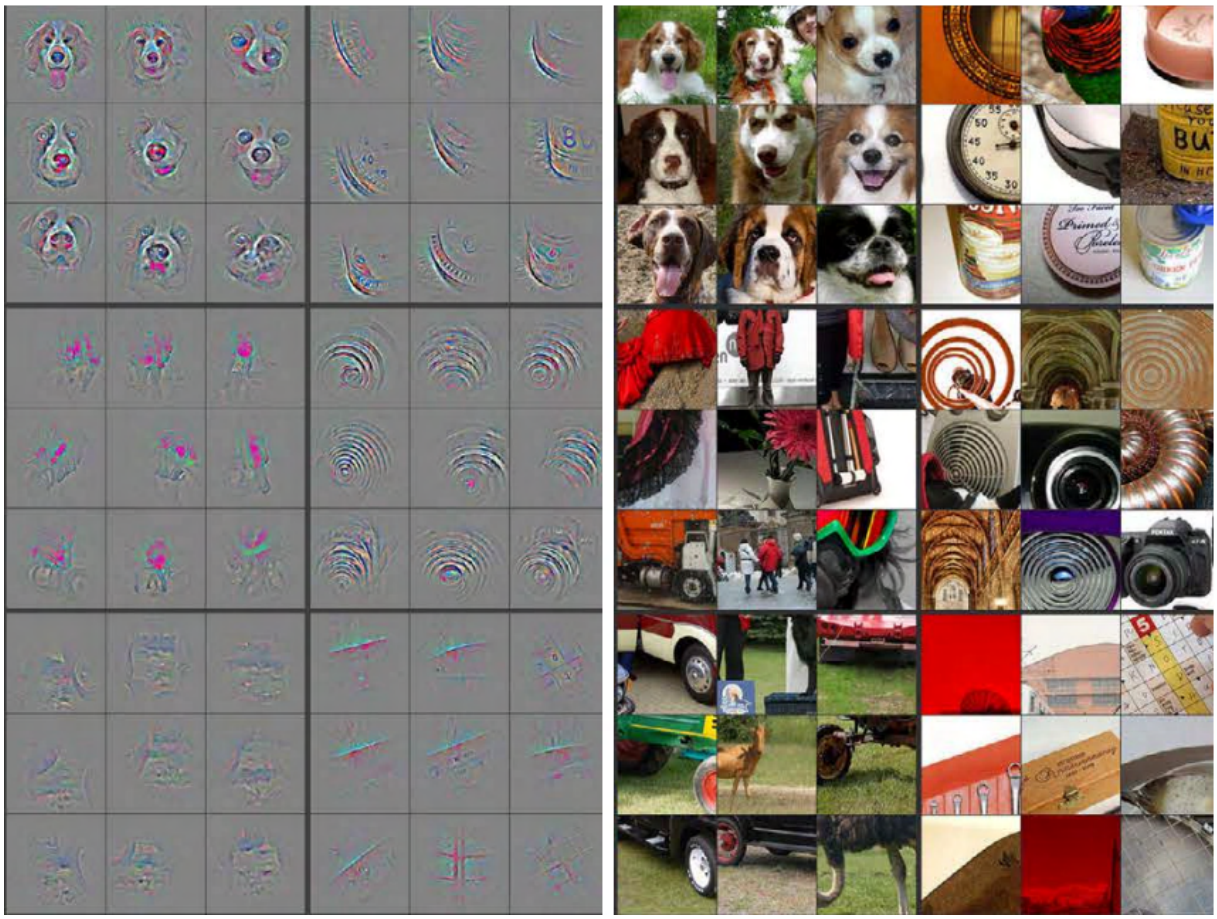


Figure 3.13: The image illustrates feature visualisations from a fully trained convolutional neural network model. The left side shows the learned filters that capture various patterns, such as edges, textures, and object parts, with higher layers focusing on more complex patterns. The right side displays images that strongly activate the corresponding filters, showing how the network detects specific visual features, such as dogs, tools, or circular objects. This figure is extracted from Zeiler et al. [Zei14], highlighting the interpretability of deep networks by visualising the learned features at different layers.

propagation from the later layer to the earlier layers. It makes the training of very much deeper neural networks possible. Some versions of ResNet have over 100 layers.

Figure 3.13 displays visualisations of features learned by a fully trained convolutional neural network, as presented by [Zei14]. The image illustrates how the network's layers progressively extract higher-level representations from the input data. The network learns to detect simple patterns such as edges, lines, and basic textures in the initial layers. As the data passes through subsequent layers, the network captures more complex structures like shapes, object parts, and, eventually, whole objects. This hierarchical feature learning demonstrates the network's ability to build sophisticated representations from simple visual elements, highlighting the effectiveness of deep learning models in understanding and interpreting visual information.

3.2.4.2 Transformers

Transformers have become increasingly popular in computer vision in recent years. Initially introduced for natural language processing tasks, transformers have since been adapted for various vision tasks, such as image classification, object detection, and segmentation.

The use of transformers was first proposed in the paper "Attention is All You Need" by Vaswani et al. in 2017 [Vas17]. The authors introduced the transformer architecture for *Natural Language Processing*, which relies solely on attention mechanisms to process input data without needing recurrent neural networks or convolutional layers. This architecture achieved state-of-the-art results on machine translation tasks and inspired researchers to explore its potential for vision tasks.

So far, NLP problems were solved using older architectures such as *Recurrent Neural Networks* or *Long Short-Term Memory* and Gated Recurrent Neural Network. These architectures raised some difficulties. When processing texts, we must transform the words into tokens to let the machine process the information. For example, for a text of 50 words, the length required to convert the content into tokens is $50 \times 1024 = 51200$ if we choose an embedding of size 1024 (a conventional size).

Vaswani et al. [Vas17] proposes to create interaction between the different positions in a sentence and compatible with sequences of various lengths. They call these operations *dot-product self-attention*. Each word under the shape of a token vector is passed by a batch of N tokens to the self-attention module and is mapped to new vectors with compatibility between them.

One of the earliest applications of transformers for vision was in the paper "Image Transformer" by Parmar et al. [Par18] in 2018. The authors proposed a transformer-based architecture for image classification that treated images as sequences of pixels and applied self-attention mechanisms to model global dependencies between pixels. While this approach achieved promising results, it was computationally expensive and required large amounts of data to train.

To address these challenges, researchers have since proposed various adaptations of the transformer architecture for vision tasks. One popular approach is to use convolutional layers to extract local features from images and then apply transformer layers to model global dependencies between these features. This approach has been used in models such as the Vision Transformer (ViT)[Dos20] and the Detecting Transformer (DETR)[Car20].

The ViT, proposed by Dosovitskiy et al. [Dos20] in 2020, treats an image as a sequence of patches and applies a transformer encoder to model global dependencies between these patches. The authors showed that the ViT achieved state-of-the-art results on several image classification benchmarks, outperforming convolutional neural networks (CNNs) on some tasks.

The DETR, proposed by Carion et al. [Car20] in 2020, is a transformer-based model for object detection that treats object detection as a set prediction problem. The DETR uses a transformer encoder-decoder architecture to model global dependencies between objects and achieves state-of-the-art results on several object detection benchmarks.

3.3 Applications of Neural Networks for Images

Image classification is one of the most common applications of neural networks for images. Image classification involves assigning a label to an input image based on its content. Convolutional neural networks (CNNs) have achieved state-of-the-art performance on various image classification benchmarks, such as ImageNet [Rus15], CIFAR-10 [Kri09], and SVHN [Net11]. CNNs use convolutional layers to extract features from the input image and pooling layers to reduce the spatial dimensions of the feature maps. The extracted features are then fed into fully connected layers for classification.

Another application of neural networks for images is object detection. Object detection involves identifying and locating objects in an image. CNNs have been used for object detection by applying a sliding window approach, where a CNN is applied to different regions of the input image to detect objects. More recently, region-based CNNs (R-CNNs) [Gir14] and their extensions, such as Fast R-CNN [Gir15] and Faster R-CNN [Ren16], have achieved state-of-the-art performance on various object detection benchmarks. These methods use a region proposal network (RPN) to generate candidate object regions and a CNN to classify and refine the proposed areas.

Semantic segmentation is another application of neural networks for images. Semantic segmentation involves assigning a label to each pixel in an image. Fully convolutional networks (FCNs) [Lon15] have been used for semantic segmentation by replacing the fully connected layers in a CNN with convolutional layers. FCNs use upsampling layers to increase the spatial resolution of the feature maps and skip connections to combine features from different layers. More recently, encoder-decoder architectures, such as U-Net [Ron15] and SegNet [Bad17], have been proposed for semantic segmentation. These architectures use an encoder network to extract features from the input image and a decoder network to generate the segmentation mask.

Neural networks have also been used to generate images. Generative adversarial networks (GANs) [Goo14] consist of two neural networks: a generator network that produces pictures and a discriminator network that evaluates the generated images. The generator network learns to create images that are indistinguishable from actual images by minimising the loss function of the discriminator network. GANs have been used for various image generation tasks, such as image-to-image translation [Iso17], super-resolution [Led17], and style transfer [Zhu17].

Face recognition is another application of neural networks for images. Face recognition involves identifying a person from a picture of their face. CNNs have been used for face recognition by extracting features from the face image and comparing them to a database of known faces. DeepFace [Tai14] and FaceNet [Sch15] are two examples of CNN-based face recognition systems that have achieved high accuracy on large-scale face recognition benchmarks.

Neural networks have also been used for medical image analysis tasks, such as tumour segmentation [Hav17], lesion detection [Jaf16], and image registration [Wu13; Zha15]. U-Net [Ron15] and its extensions, such as V-Net [Mil16] and 3D U-Net [Çiç16], have been widely used for medical image segmentation. These architectures use skip connections to combine features from different layers and have achieved state-of-the-art performance on various medical image

segmentation benchmarks.

In conclusion, neural networks have great potential in various image processing and computer vision tasks. In this section, we have explored some of the most common applications of neural networks for images, including image classification, object detection, semantic segmentation, image generation, face recognition, and medical image analysis. With the increasing availability of large-scale labelled image datasets and advances in neural network architectures and training methods, neural networks' performance on these tasks is expected to improve.

3.4 Conclusion

In this chapter, we have introduced deep learning and its applications. We started with supervised learning, explaining how models can learn from labelled data to make predictions. We then discussed the difference between regression and classification tasks, which are important in many fields.

Next, we introduced neural networks inspired by the human brain. We explained how they are composed of layers of neurons and can learn complex patterns in data. We also described the backpropagation algorithm, which is essential for training neural networks by adjusting the weights to minimise the loss function.

We also discussed different types of neural networks, such as Convolutional Neural Networks (CNNs) and Transformers. CNNs are very effective for image processing tasks because they can capture spatial features in images. Transformers are a newer type of network that uses attention mechanisms and are becoming popular in both natural language processing and computer vision.

Finally, we looked at some applications of neural networks in image analysis, like image classification, object detection, and semantic segmentation. These applications show how powerful deep learning can be in solving complex problems.

This chapter provides the technical background needed to understand the methods used in this thesis. With this foundation, we can now proceed to explore more advanced topics in the following chapters.

Chapter 4

Chan-Vese Attention U-Net: An Attention Mechanism for Robust Segmentation.

Objectifs

This chapter addresses the problem of object segmentation. Segmentation is a critical task in image analysis, particularly in applications requiring precise delineation of objects. When studying the results of a segmentation algorithm using cnn, one wonders how robust the results are. This leads to questioning the possibility of using such an algorithm in applications without room for doubt. In this chapter, we present a new attention gate based on Chan-Vese energy minimisation, which uses a standard CNN architecture such as the U-Net model to control the segmentation masks more precisely. This mechanism allows us to obtain a constraint on the segmentation based on the resolution of a PDE and allows the gradient to propagate through the optimisation process. The study of the results allows us to observe the spatial information retained by the neural network on the region of interest and obtain competitive results on the binary segmentation. We illustrate the efficiency of this approach for medical image segmentation on a database of brain images.

Contents

4.1 Introduction

This is a joint work with Laurent D. Cohen. Accepted at GSI 2023 conference, it has been published online as part of the Proceedings of the 6th International Conference on Geometric Science of Information (GSI 2023).

Literature Review

Medical image segmentation is a crucial task that requires significant time and effort from medical experts. Although various solutions, including Convolutional Neural Networks (CNNs), have been proposed to automate this process, the need for efficient and reliable methods still exists. While CNNs have shown promising results in medical image segmentation, their opaque reasoning and the sensitive nature of medical data raise concerns regarding their applicability in real-world hospital settings, especially for medical staff who need to be trained in machine learning. Researchers have explored integrating geometric or topological properties into neural networks to address these challenges and incorporate information beyond adjacent pixels for segmentation tasks.

Convolutional neural networks have revolutionised image classification and segmentation, with architectures like the fully convolutional network by Long et al. [Lon15] and the U-Net by Ronneberger et al. [Ron15] standing out for their performance and versatility. These architectures have been extensively tested on various applications, such as MRI segmentation of the brain [Kle16] and heart [Pop18], as well as CT scans of thoracic organs [Ger19]. Numerous modifications have been made to improve the efficiency of these complex structures, but challenges still need to be solved in achieving precise boundary delineation and incorporating domain-specific knowledge.

One avenue of research has focused on combining CNNs with Active Contour Models (ACMs) to leverage the strengths of both methods. One of the first papers on integrating neural networks and active contours is by Rupprecht et al. [Rup16], who proposed integrating a CNN with an ACM by training a class-specific CNN to predict a vector field that guides contour evolution (See Figure 4.1). The method involves extracting small patches from the evolving contour and predicting vectors pointing towards the nearest object boundary. The predicted vector field is used to evolve the contour within a Sobolev space, as proposed by Sundaramoorthi et al. [Sun07], ensuring smooth and robust contour evolution while mitigating the impact of spurious local predictions.

Soon after, another idea was proposed by Wang et al. [Wan18] also to mitigate user information. The authors proposed a novel integration of Conditional Random Fields (CRFs) with Convolutional Neural Networks (CNNs) for interactive image segmentation. The method leverages CRFs to model spatial dependencies with unary and pairwise potentials, where the unary potentials are derived from CNN outputs, and the pairwise potentials are represented by a flexible neural network called Pairwise-Net. This allows for learning freeform pairwise functions rather than using fixed Gaussian functions, thereby enhancing the modelling of complex relationships. The CRF is implemented as a Recurrent Neural Network (RNN) to facilitate end-

to-end training with the CNNs. The iterative mean-field approximation method minimises the Gibbs energy, updating the distribution to minimise the KL divergence. A key innovation is incorporating user interactions as hard constraints within the CRF framework, using geodesic distance transforms to convert user-provided scribbles into features. This ensures the segmentation respects user inputs by setting the probabilities for user-labeled pixels accordingly. To maintain computational efficiency, the pairwise connections are restricted to local patches, reducing complexity and mitigating long-distance dependency issues that could corrupt segmentation in medical images with low contrast. The Pairwise-Net is pre-trained using a synthetic training set.

Building on this idea, Marcos et al. [Mar18] proposed "Learning Deep Structured Active Contours End-to-End," aiming to enhance the precision of building footprint segmentation by integrating the geometric constraints inherent in ACMs with the robust feature learning capabilities of CNNs (See Figure 4.2). Compared to Rupprecht et al. [Rup16], this work is one of the first to allow the gradient of the ACM to update the weights of the CNN. The CNN predicts the parameters of the ACM energy function, including data terms, curvature penalisation, and balloon terms, guiding the ACM to fit object boundaries accurately. Training involves a structured prediction problem using a Structured Support Vector Machine (SSVM) loss, allowing for end-to-end training. The SSVM loss function is convex but not necessarily differentiable. Hence, the optimisation employs subgradient methods to find the most violated constraints and update the network parameters through backpropagation.

In the medical domain, Hatamizadeh et al. [Hat19] proposed the "Deep Active Lesion Segmentation" (DALS) framework (See Figure 4.3), integrating CNNs and ACMs to enhance lesion segmentation in medical images. The method involves a CNN that predicts an initial probability map and local parameter maps to guide the ACM's contour evolution. The ACM refines the initial segmentation by evolving the contour to minimise an energy functional influenced by these parameter maps. A structured loss, typically a Dice coefficient, measures the final segmentation accuracy and backpropagates through the CNN to update its weights, indirectly accounting for the ACM evolution. However, a limitation is that the CNN may focus on generating a perfect initial segmentation, potentially rendering the ACM step redundant. The ACM provides a regularisation effect during early training stages rather than contributing directly to the final segmentation accuracy.

Similarly, Akbarimoghaddam et al. [Akb22] introduced an innovative image segmentation method that integrates CNNs with ACMs. The authors propose a Locally Controlled Distance Vector Flow (LCDVF) to enhance the ACM's effectiveness, leveraging CNN-predicted initialisation and parameter maps for improved performance. The framework employs a dual CNN architecture trained simultaneously: one CNN predicts the internal energy parameters of the ACM and the balloon force, inspired by the DSAC approach. At the same time, the other CNN generates a ground truth mask used to derive the initialisation circle and distance transform. The ACM evolves a contour represented as a set of polygon points by minimising an energy functional that combines external energy with spatially varying parameters predicted by the CNN. This approach enhances capture range and reduces sensitivity to the initial contour location.

The proposed DALS architecture. DALS is a fully automatic framework that does not re-

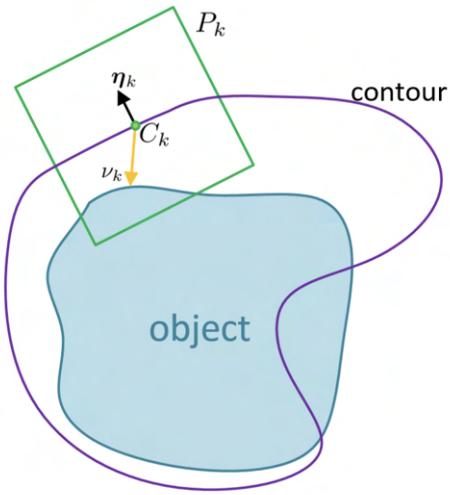


Figure 4.1: The diagram illustrates the concept of combining Convolutional Neural Networks (CNNs) with Active Contour Models (ACMs) to guide contour evolution towards object boundaries. The contour is evolved based on a vector field predicted by a CNN, where each vector points towards the nearest boundary of the object. C_k represents a point where a small patch P_k is extracted, with the normal vector ν_k pointing towards the object boundary and the tangential vector η_k maintaining contour smoothness.(Figure from [Rup16]

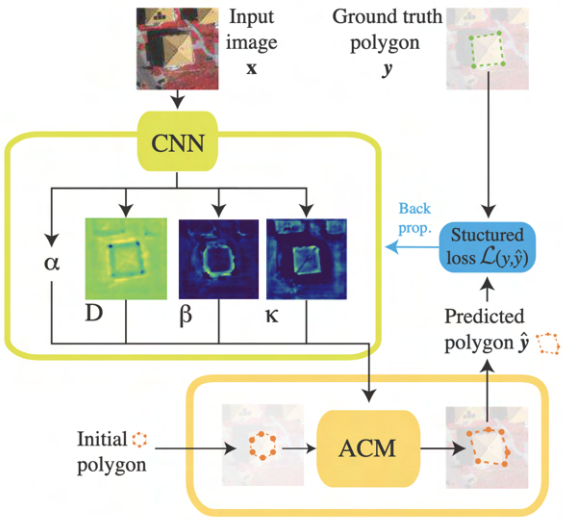


Figure 4.2: DSAC idea. The CNN predicts the values of the energy terms to be used by the active contour model (ACM): a global α for the length penalisation and maps for local D, the data term, β, the curvature penalisation and κ, the balloon term. After ACM inference, a structured loss is computed and given to the CNN, whose parameters can then be updated using backpropagation. (Illustration from [Mar18])

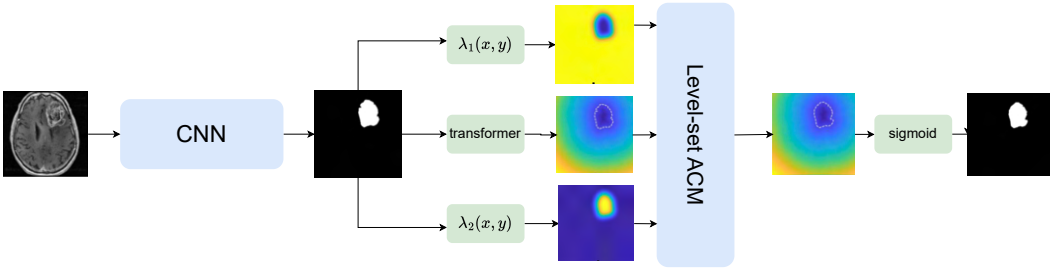


Figure 4.3: The proposed DALS architecture. DALS is a fully automatic framework without the need for human supervision. The CNN initialises and guides the ACM by learning local weighted parameters.

quire human supervision. The CNN initialises and guides the ACM by learning local weighted parameters. The methodology proposed in the paper from Chen et al. [Che19] integrates traditional Active Contour Models (ACMs) with Convolutional Neural Networks (CNNs) to improve segmentation accuracy by incorporating geometric constraints directly into the loss function. The framework introduces a novel Active Contour Loss (AC Loss) function that combines boundary length and region-based terms to ensure precise and smooth segmentation boundaries. The AC Loss function consists of two components: the Length term, which penalises the length of the predicted segmentation boundary to encourage smoother contours, and the Region term, which measures the difference in intensity inside and outside the predicted contour against the ground truth, ensuring accurate region delineation. The Length term is calculated as the integral of the gradient magnitude of the predicted segmentation. In contrast, the Region term is based on the mean intensities inside and outside the contour.

To address issues related to imbalanced class data and weak boundary delineation in medical images, Le et al. [Le21] proposed a two-branch deep network architecture. The first branch employs a conventional encoder-decoder network, such as a U-Net or Fully Convolutional Network (FCN), to extract high-level semantic features and generate a coarse segmentation map. The second branch, the Narrow Band Active Contour (NB-AC) attention model, focuses on refining segmentation boundaries by concentrating on lower-level features around the object contours. This model operates within a narrow band around the predicted contour, treating it as a hyperplane and using the data within it to adjust its position and orientation. The NB-AC model incorporates contour length and regional energy terms to enhance boundary precision, guiding the contour to minimise boundary length while accurately fitting the region.

Furthermore, Wu et al. [Wu20] proposed the Deep Parametric Active Contours (DPAC) framework, integrating CNNs with ACMs to enhance boundary precision and segmentation accuracy. The CNN predicts local weighted parameter maps that guide the ACM's evolution by controlling the contour's internal and external energy terms. The internal energy, influenced

by elasticity and bending rigidity maps, penalises the curve's length and curvature to maintain smoothness. The external energy, driven by Gradient Vector Flow (GVF) and normal force maps, directs the curve towards object boundaries. The DPAC framework is trained end-to-end using backpropagation, adjusting the predicted parameter maps iteratively to evolve the curve towards accurate boundaries in medical images.

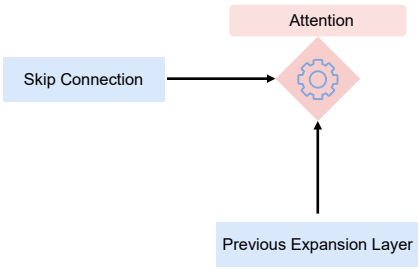
In addition to these methods, other attempts have been made to integrate geometric or topological properties into neural networks. For instance, Zhang et al. [Zha20] presented a model where the neural network predicts the parameters for initialising the active contour model and an initial contour. Learning is achieved by combining the error produced by the neural network with that from the active contour usage. Ma et al. [Ma20] proposed a fully integrated geodesic active contour model, where the neural network learns to minimise the energy functional of the model. This encoder-decoder network outputs a contour map instead of a probability map for segmentation, based on the active contour method proposed by Caselles et al. [Cas97].

Despite these advancements, challenges remain due to CNNs' opaque reasoning and the sensitive nature of medical data, which raise concerns regarding their applicability in clinical settings. With our proposed attention gate mechanism, we aim to study how neural networks behave to produce the desired output segmentation, providing more transparency and interpretability.

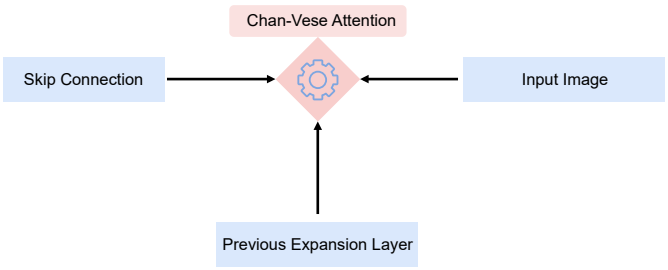
The rest of this chapter is organised as follows. In Section 4.2, we introduce our experimental procedure for the Chan-Vese Attention Gate. In Section 4.3, we present the main results of our experiments and provide a discussion of our work.

This chapter introduces a novel hybrid approach combining classical segmentation techniques based on functional energy minimisation with deep learning. Our method features a new attention gate, the *Chan-Vese Attention Gate*, which integrates information from the level sets method of the well-established Chan-Vese functional[Cha01]. In the proposed segmentation framework, we integrate attention mechanisms within a standard convolutional neural network architecture, enhancing the precision of region segmentation. Figure 4.4a illustrates the use of skip connections and attention mechanisms. In contrast, Figure 4.4b demonstrates incorporating the Chan-Vese model into the architecture to impose additional shape constraints on the segmentation results. Unlike traditional deep learning methods that rely solely on the neural network to improve image segmentation, our approach leverages resolution information to achieve more accurate results.

To demonstrate the effectiveness of our method, we conducted comprehensive experiments on the TCGA LGG database [Ped]. Given the sensitive nature of medical image segmentation, it is crucial to ensure the validity of our results. Our approach achieved at least equivalent results to previous networks while remaining simple to optimise, with only slightly longer computation time. Our approach represents a significant advancement in medical image segmentation, offering a more accurate and efficient solution for this critical field.



(a)



(b)

Figure 4.4: (a) Visualization of the network architecture with skip connections and the previous expansion layer integrated with an attention mechanism. The attention mechanism is used to refine the segmentation masks by focusing on regions of interest, improving precision. (b) Diagram illustrating the input image, skip connection, previous expansion layer, and the integration of Chan-Vese attention. This architecture enhances segmentation performance by leveraging the Chan-Vese model to impose shape constraints on the segmentation process.

4.2 Methodology

4.2.1 The U-Net architecture

The U-Net architecture [Ron15], a convolutional neural network, is widely used for medical image segmentation. This process involves separating the regions of interest from the rest of the image. This architecture is particularly effective because it maintains the structure of an image during the transformation process, which involves converting an image into a vector, processing it, and then converting it back into an image. This is achieved by using features extracted during the contraction phase, also known as the encoding or downsampling path, which helps to preserve important spatial information.

The U-Net architecture takes the shape of a "U", hence its name, and comprises three main parts: contraction, transition, and expansion. The contraction path, the left side of the "U", applies several blocks, each containing convolution and pooling layers. These layers work together to capture the context in the image, with the number of feature maps doubling at each stage, thereby increasing the depth of the network and enabling it to learn more complex features.

The transition part, the bottleneck, uses convolution layers to create a compact image representation, which helps reduce the network's computational complexity. The expansion path (which is the right side of the "U") uses a combination of convolution (See Section 3.2.4.1) and up-sampling layers. These layers work together to recover the spatial information lost during the contraction phase and to reconstruct the image. The number of expansion blocks is the same as the number of contraction blocks, ensuring a symmetrical architecture that balances the encoding and decoding of the image.

Finally, the network's output is obtained through a last convolutional layer, which applies a 1x1 convolution to map each component feature vector to the desired number of classes. This final layer provides the segmentation map, highlighting the regions of interest in the image, thereby completing the image segmentation process.

The U-Net architecture is particularly effective for medical image segmentation due to several key features distinguishing it from other convolutional neural networks.

Firstly, U-Net uses many feature channels (See Section 3.2.4.1), up to 1024 in the original implementation, allowing it to capture more contextual information and learn more complex features. This is especially important in medical image segmentation, where the regions of interest can be small and intricate, and their differences can be subtle (See Figure 4.6).

Secondly, U-Net uses skip connections, or shortcut connections, to concatenate the feature maps from the contraction path to the corresponding feature maps in the expansion path. These connections help preserve the spatial information lost during the pooling operations in the contraction path. By combining the high-resolution features from the contraction path with the up-sampled features from the expansion path, the network can make more accurate predictions and produce more precise segmentation masks.

Thirdly, U-Net uses a symmetric architecture, with the number of expansion blocks being the same as the number of contraction blocks. This symmetry helps to balance the encoding and decoding of the image, ensuring that the network does not lose too much information

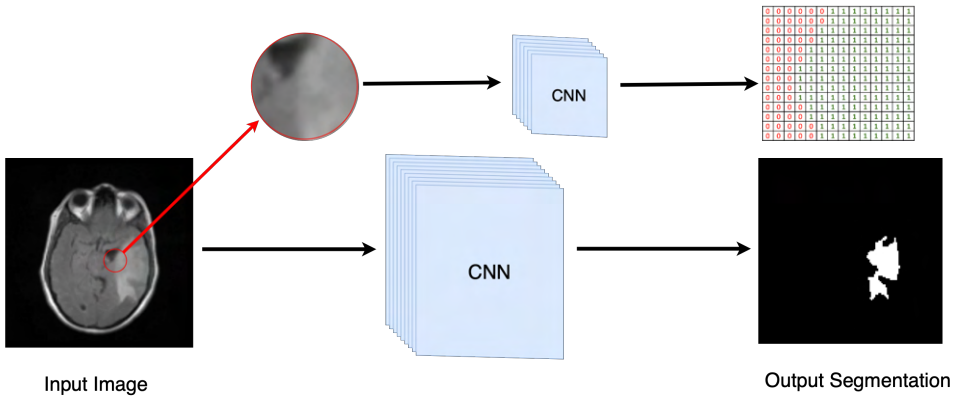


Figure 4.5: The diagram illustrates the process of brain tumour segmentation using a convolutional neural network (CNN). The brain's input image, an MRI scan, is processed by a CNN to extract key features, emphasising a specific region of interest (ROI) for detailed analysis. The CNN produces both a feature map, representing the learned features of the input, and the final binary output segmentation, where the white region highlights the segmented tumour area.

during the downsampling process and can recover the details during the upsampling process.

Finally, U-Net uses a pixel-wise softmax function (See Equation3.9) in the final layer, which allows it to output a probability map for each class. This is particularly useful in medical image segmentation, where the output is often a binary mask indicating the presence or absence of a particular structure (See Figure 4.5). U-Net can provide more nuanced predictions by outputting a probability map, indicating the network's confidence level for each pixel.

In summary, the U-Net architecture works best for medical image segmentation because it captures more contextual information, preserves spatial information through skip connections, balances the encoding and decoding of the image through a symmetric architecture, provides nuanced predictions through a pixel-wise softmax function, and learns a global representation of the image through end-to-end training.

In this chapter, we have based our study on the architecture of the U-Net [Ron15], which is probably the most widely used CNN in medical image segmentation.

4.2.2 Attention Gate in U-Net architecture

This section explores the concept of attention within the UNet architecture. Attention, a fundamental element in various deep learning applications, is illustrated in Figure 4.7, emphasising salient features while diminishing irrelevant background noise. Specifically, in the context of UNet, attention mechanisms focus on significant objects, such as sheep, rather than the background, thereby enhancing the model's efficiency by concentrating computational resources on pertinent areas. This selective emphasis aids in better generalisation of the network without necessarily increasing its compute time. The quality of results obtained from integrating attention into UNet will be examined in subsequent sections, where we will determine whether this addition provides a marginal or substantial improvement over the standard UNet configura-

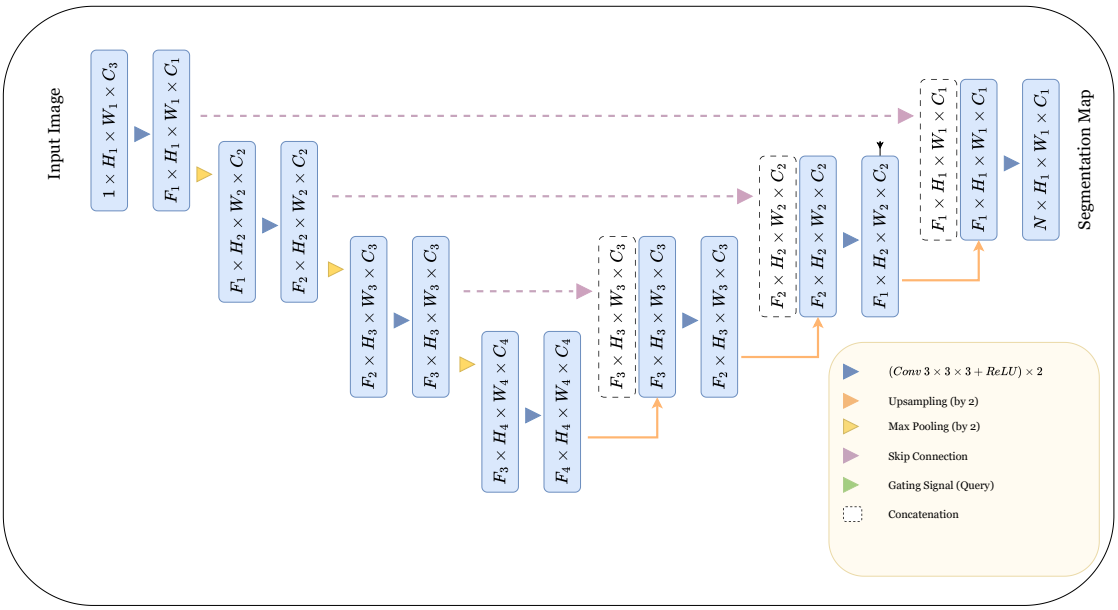


Figure 4.6: A block diagram of the U-Net segmentation model. The input image is progressively filtered and downsampled by a factor of 2 at each scale in the encoding part of the network. N denotes the number of classes.

tion.

There are two primary types of attention mechanisms used within neural networks: hard attention and soft attention.

Hard attention is characterised by explicitly localising relevant regions within an image by segmenting specific areas of interest, such as sheep. These segmented regions are subsequently processed independently through the convolutional network. However, hard attention employs discrete, non-differentiable operations, rendering standard backpropagation inapplicable. Consequently, alternative learning strategies, such as reinforcement learning techniques or Monte Carlo methods, are necessitated to optimise the region selection process. This introduces computational complexity and increased resource requirements.

In contrast, soft attention operates by assigning continuous, differentiable weights to various parts of the image, thereby enabling the application of backpropagation. This mechanism computes a weighted sum of the input features, with the weights learned and refined during training. Soft attention empowers the model to dynamically adjust its focus based on the relevance of different regions, thereby enhancing feature extraction and overall network performance. The training involves gradually optimising these weights via gradient descent, allowing the model to concentrate on pertinent areas as training epochs increase progressively.

The implementation of soft attention typically involves the following steps:

1. Linear transformation of the input features into query, key, and value vectors;

2. Computation of attention scores by comparing the query vectors with the key vectors,

Figure 4.7: The image illustrates a sheep being highlighted by an attention mechanism within a deep learning model. The attention map focuses on the key features of the sheep, such as its body and head, emphasising the regions most relevant for the model's decision-making process. This visual representation shows how the attention mechanism directs the model's focus, enabling more accurate identification and segmentation of the sheep in the image.

often using dot-product or scaled dot-product;

3. Normalisation of the attention scores, typically using the softmax function, to produce attention weights;

4. Generation of context vectors, which emphasise essential features, by weighting the value vectors by the attention weights.

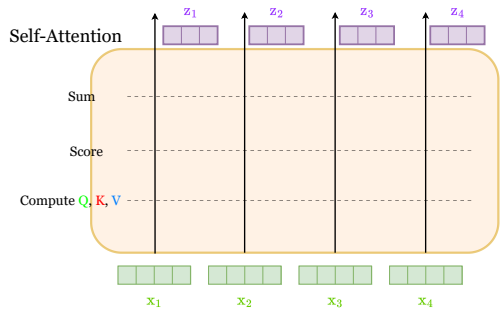
Soft attention is computationally efficient and integrates seamlessly for various Deep Learning tasks. This flexibility makes it particularly suitable for complex tasks such as image captioning, machine translation, and visual question answering, where discerning and prioritising specific parts of the input data is critical.

In the case of transformers the self-attention module works as follows: imagine you have a sentence broken into four tokens (words or subwords) and want to understand the connection between these tokens. Self-attention helps the model do this by processing each token with respect to all the others.

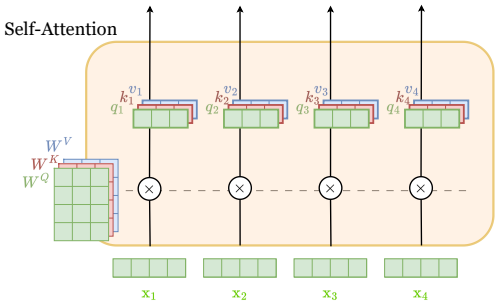
The process begins by creating three vectors for each token: Query (Q), Key (K), and Value (V) vectors. The query represents the current token, and the key vector represents all the other tokens it is compared against. For each token, its query is compared with the keys of every other token (including itself), and this comparison gives a score that tells us how much attention should be paid to the other tokens. Figure 4.8[1] graphically presents the self-attention module's different steps for better understanding in the case of Natural Language Processing, where the input is a series of tokens.

Once we have these scores, we use them to weigh the value vectors (which contain the information of the tokens) and sum them up. Tokens with higher scores will contribute more

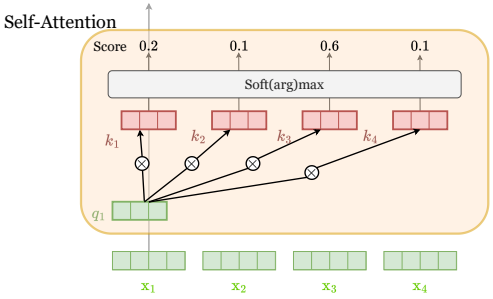
[1]Illustrations inspired by the blog post of Jay Alammar on GPT2. https://jalammar.github.io/illustrated-gpt2/



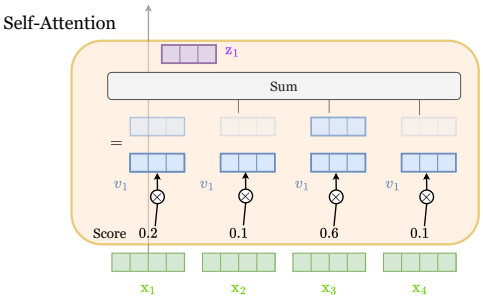
(a) An overview of self-attention computation, illustrating the process of generating Query (Q), Key (K), and Value (V) vectors from input tokens, followed by calculating the attention scores.



(b) Illustration of the process to create the query, key and value vectors for each input token by multiplying by weight matrices W^Q, W^K, W^V.



(c) Depiction of how the query vector of one token is compared against the key vectors of other tokens to compute attention scores in the self-attention mechanism. Illustrating the use of the softmax function to normalise attention scores, making them interpretable as probabilities in the self-attention process.



(d) Visualisation of how attention scores are used to weight the value vectors and produce a final output that captures the contextual relationships between tokens.

Figure 4.8: The images provide a step-by-step visualisation of the self-attention mechanism in a neural network. Each subfigure illustrates different stages, from generating query, key, and value vectors (a, b) to computing and normalising attention scores (c) and finally producing the output by weighting the value vectors based on these scores (d).

to the final result, capturing the essential context. After applying this self-attention process for all tokens, the model better understands each token in relation to others, which is then passed to the next layer in the network.

Recent advancements in attention mechanisms have led to the development of self-attention and the Transformer architecture, which extend the principles of soft attention to capture long-range dependencies and contextual relationships within data. In self-attention mechanisms, each input sequence element attends to all other elements, enabling the model to construct a global context and significantly improve performance on tasks involving sequential data. Using multi-head self-attention, the Transformer architecture further enhances this capability by allowing the model to simultaneously focus on different aspects of the input.

To recall, the architecture of a standard U-Net is characterised by repetitive convolutional blocks and prominent skip connections. The skip connections play a vital role in maintaining spatial information that is progressively downsampled in the encoder path and later upsampled in the decoder path. This retained spatial information offers essential context that facilitates output reconstruction during upsampling. However, a challenge emerges due to the relatively weak feature representations in the early stages of the encoder path, as these layers primarily capture rudimentary features.

Soft attention mechanisms can be incorporated into the skip connections to address this limitation. By introducing attention gates at these junctions, weights can be assigned to specific regions of interest, such as mitochondria in an image. These attention gates amplify the model's focus on relevant features, enhancing overall feature representation.

In a U-Net augmented with attention mechanisms, each skip connection is equipped with an attention gate, comprising two main components: the gating signal (query) and the input from the skip connection. The gating signal, originating from deeper layers, contains rich feature information, while the input from the skip connection offers spatial information from earlier layers. The attention mechanism aligns these inputs, generating attention coefficients highlighting significant features.

The efficacy of this approach is demonstrated by the progressively refined attention coefficients over multiple training epochs. These coefficients become increasingly concentrated on the regions of interest as training advances, illustrating the model's enhanced capacity to prioritise relevant features.

To understand the attention gate within the U-Net architecture (See Figure 4.9), it is essential to analyse its constituent components and operations. The attention gate primarily takes two inputs: x and g. The input g denotes the gating signal, typically derived from a deeper layer in the network, and thus contains rich feature representations. In contrast, x represents the input from an earlier layer, which carries detailed spatial information but weaker feature representations.

Due to their respective positions in the network, the gating signal g and the input x often have different dimensionalities. Both inputs undergo a 1×1 convolution to address this. For g, this convolution preserves its original dimensions, while for x, a stride of 2×2 is applied, effectively reducing its spatial resolution by half. After the convolution, both tensors are reshaped to a matching dimension of $64 \times 64 \times 128$, where 128 signifies the number of filters.

The reshaped tensors are then added element-wise, taking advantage of the fact that aligned

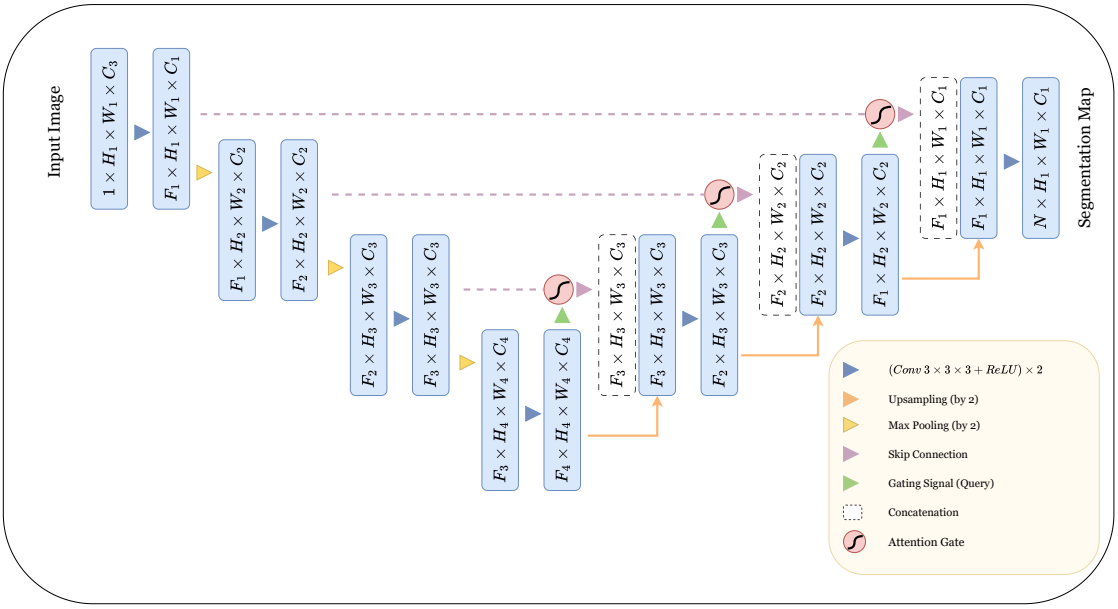


Figure 4.9: A block diagram of the Attention U-Net segmentation model. The input image is progressively filtered and downsampled by a factor of 2 at each scale in the encoding part of the network. N denotes the number of classes. Attention gates (AGs) filter the features propagated through the skip connections. Schematic of the AGs is shown in Figure 4.10. Feature selectivity in AGs is achieved by use of contextual information (gating) extracted in coarser scales. (Figure inspired by [Okt18]

weights (e.g., 0.9 and 0.9) result in larger summed values (e.g., 1.8), while unaligned weights (e.g., 0.1 and 0.1) yield relatively smaller sums (e.g., 0.2). This additional step accentuates significant feature alignments. The resulting tensor is passed through a ReLU activation function, eliminating negative values and maintaining positive ones linearly.

Next, a 1×1 convolution is applied to the activated tensor, generating a single-channel output of size $64 \times 64 \times 1$. These values, essentially the attention weights, are then normalised to a [0,1] range via a sigmoid function. To match the original dimensions of x, these weights are upsampled to 128×128.

The final step entails element-wise multiplication of the upsampled weights with the input x, adjusting each pixel value in x based on its corresponding weight. This process dynamically enhances the significant regions in x during training, as these weights are refined iteratively through backpropagation.

To implement this in code, the attention block can be structured as follows:

1. **Convolution Operations**:
    - $\theta(x)$: 1×1 convolution with a stride of 2×2 applied to x.
    - $\phi(g)$: 1×1 convolution applied to the gating signal g.
2. **Addition and Activation**:
    - The outputs of $\theta(x)$ and $\phi(g)$ are added element-wise and passed through a ReLU activation.
3. **Attention Coefficients**:
    - Apply a 1×1 convolution with a single filter to the activated sum to produce the attention coefficients.
    - Normalise these coefficients using a sigmoid function.
4. **Upsampling**:
    - Upsample the normalised coefficients to the original dimensions of x.
5. **Element-wise Multiplication**:
    - Multiply the upsampled coefficients with x element-wise.

Here is a brief implementation:

```
def attention_gate(x, g):
    theta_x = Conv2D(128, (1, 1), strides=(2, 2))(x)
    phi_g = Conv2D(128, (1, 1))(g)

    add_xg = Add()([theta_x, phi_g])
    relu_xg = Activation('relu')(add_xg)

    psi = Conv2D(1, (1, 1))(relu_xg)
    sigmoid_xg = Activation('sigmoid')(psi)
```

```
    upsampled_psi = UpSampling2D(size=(2, 2))(sigmoid_xg)

    y = Multiply()([upsampled_psi, x])

    return y
```

This function aligns x and g, computes the attention coefficients, normalises them, and applies them to the input x. The attention gate thus refines feature representations, enhancing the U-Net's ability to focus on significant features. As demonstrated in subsequent images from the referenced paper, the attention coefficients become increasingly focused on the regions of interest as training progresses, illustrating the effectiveness of this method.

4.2.3 Chan-Vese Energy Minimization

We briefly recall the Chan-Vese method, presented in 2.3.2, used to segment a binary image. Let I be the given grayscale image on a domain Ω to be segmented. The Chan Vese method looks for a piece-wise constant approximation of an image where two regions are separated by an unknown boundary curve C. This is obtained through the minimisation of the following energy depending on curve C and the constant values c1 and c2 inside and outside the curve:

$$E(C, c_1, c_2) = \mu \times \text{Length}(C) + \nu \times \text{Area}(inside(C)) + \lambda_1 \int_{inside(C)} |u_0(x,y) - c_1|^2 \mathrm{d}x\mathrm{d}y + \lambda_2 \int_{outside(C)} |u_0(x,y) - c_2|^2 \mathrm{d}x\mathrm{d}y, \quad (4.1)$$

where $\mu, \lambda_1, \lambda_2$ are positive constants.

Energy minimisation is simplified by replacing the curve C with a level set function ϕ. The inside region is then the set where $\phi > 0$ and the outside region the set where $\phi < 0$. With the help of the Heavyside function H, the energy becomes:

$$F(c_1, c_2, \phi) = \mu \int_\Omega \delta(\phi(x))|\nabla\phi(x)|dx + \nu \int_\Omega H(\phi(x))dx + \lambda_1 \int_\Omega |I(x) - c_1|^2 H(\phi(x))dx + \lambda_2 \int_\Omega |I(x) - c_2|^2 (1 - H(\phi(x)))dx, \quad (4.2)$$

where the term following μ represent the length of the contour, ν the area inside the contour and δa the Dirac mass. This is useful since now all integrals are on the whole domain Ω, and differentiation is thus made simpler.

$$(P) : \arg\min_{c_1, c_2, \phi} F(c_1, c_2, \phi) \quad (4.3)$$

Minimisation is solved using the associated Euler-Lagrange Equation that evolves ϕ instead of evolving directly curve C.

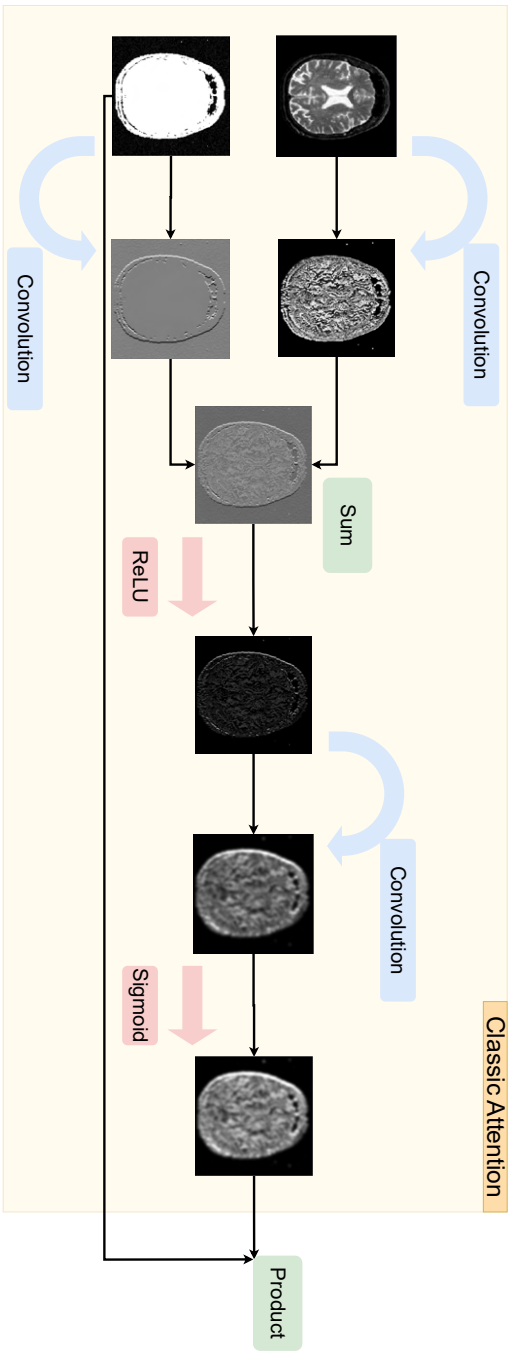


Figure 4.10: The diagram illustrates the classic attention process in which feature maps undergo convolution, followed by element-wise operations (sum, product), and are modulated by a sigmoid activation to highlight important regions before being passed through further layers.

4.2.4 Chan-Vese Attention in U-Net architecture

We propose an enhanced attention mechanism integrated into the U-Net architecture to improve the accuracy of medical image segmentation. Building upon the attention method introduced by Oktay et al. [Okt18], which highlights informative regions in skip connections, our approach incorporates information from the Chan-Vese method to perform intermediate segmentation at each block of the expansion phase.

As illustrated in Figures 4.10 and 4.11, the feature representation from the skip connection at layer l, denoted as $X_i^l \in \mathbb{R}^{Fl}$ for spatial position i, is combined with a gating vector $g_i \in \mathbb{R}^{F_g}$, which contains contextual information from the previous layer. Following the additive attention formulation, the attention coefficients $\alpha_i^l \in [0,1]$ are computed as:

$$q_{\text{att}}^l = \psi^\top \left(\sigma_1 \left(W_x^\top X_i^l + W_g^\top g_i + b_g\right)\right) + b_\psi, \tag{4.4}$$

$$\alpha_i^l = \sigma_2 \left(q_{\text{att}}^l\right), \tag{4.5}$$

where σ_1 is the ReLU activation function, $\sigma_2(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function, $W_x \in \mathbb{R}^{F_l \times F_{\text{int}}}$ and $W_g \in \mathbb{R}^{F_g \times F_{\text{int}}}$ are weight matrices, $b_g \in \mathbb{R}^{F_{\text{int}}}$ and $b_\psi \in \mathbb{R}$ are bias terms, $\psi \in \mathbb{R}^{F_{\text{int}}}$ is a weight vector, and F_{int} is the number of intermediate features. The attention coefficients modulate the feature map by element-wise multiplication: $\hat{X}_i^l = X_i^l \cdot \alpha_i^l$, allowing the network to focus on salient regions relevant to the segmentation task.

To refine the attention mechanism, we apply a differentiable distance transform D (See Section 2.5) to the attention coefficients, obtaining a transformed attention map:

$$\beta_i^l = D(\alpha_i^l) = -\lambda \log \left(\alpha_i^l * \exp\left(-\frac{d(\cdot,0)}{\lambda}\right)\right), \tag{4.6}$$

where $d(\cdot,0)$ is the Euclidean distance to the zero level-set, $*$ denotes convolution, and λ is a scaling parameter. This transformed map β_i^l serves as the initial contour for the Chan-Vese segmentation.

In addition to the main branch, we introduce a secondary branch to emphasise the tumorous regions of the input image $I \in \mathbb{R}^{N \times H \times W \times C}$, where N is the batch size, H and W are the height and width, and C is the number of channels. The input image is resized to match the dimensions of layer l and transformed as:

$$\gamma_i^l = \sigma_2 \left(W_{x'}^\top \left(W_{x'}^\top X_i^l + \sigma_2(I_i)\right) + b_W\right), \tag{4.7}$$

where $W_{x'}$ is the weight matrix for the 1×1 convolution, b_W is a bias term, and I_i is the input image at position i. The transformation γ_i^l reduces the intensity in regions far from the tumour, mitigating the inclusion of undesirable areas in the segmentation.

Finally, the segmentation mask and refined attention coefficient ζ_i^l are obtained by solving the Chan-Vese problem:

$$\zeta_i^l = \text{CV}(\gamma_i^l, \beta_i^l, \mu, \nu), \tag{4.8}$$

where CV denotes the Chan-Vese segmentation function, and μ and ν are positive regularisation parameters controlling the smoothness and fidelity of the segmentation. The function CV iteratively segments the image γ_i^l using β_i^l as the initial contour, enhancing the network's ability to focus on the most relevant features.

The integration of the Chan-Vese method into our attention mechanism offers several benefits:

1. **Intermediate Segmentation:** It enables the network to perform intermediate segmentation at each block of the expansion phase, improving the final segmentation accuracy.
2. **Facilitated Learning:** It provides a control signal that facilitates learning and enhances convergence during training.
3. **Focused Attention:** By emphasising the tumorous regions, the network improves its focus on critical areas in medical images.

In summary, our proposed attention method extends the mechanism by Oktay et al. [Okt18] by incorporating the Chan-Vese method and a secondary branch emphasising tumorous regions. This approach refines the attention maps, allowing the network to concentrate on the most relevant features while ignoring irrelevant ones, thereby enhancing the accuracy of medical image segmentation.

4.2.5 Differentiability of the Optimisation Problem

In the proposed model, we integrate the Chan-Vese method within an attention mechanism to enhance segmentation in a U-Net-like architecture for medical imaging. This approach raises differentiability and optimisation challenges, especially when using non-differentiable components such as distance transforms and algorithms like the Chan-Vese segmentation.

The central issue in this integration is the differentiability of the Chan-Vese method and its reliance on the distance transform. Traditionally, the Euclidean Distance Transform (EDT) and the Chan-Vese model are not directly differentiable. This poses a problem in end-to-end neural network training, where gradients are required for backpropagation.

To solve the issue with the non-differentiability of the Euclidean Distance Transform usual implementation, we use a differentiable variant of the Euclidean Distance Transform, as proposed by Pham et al. [Pha21]. The distance transform, D_x, used in our attention block takes a feature map and computes the distance of each pixel to the nearest boundary, ensuring that this operation remains compatible with gradient-based optimisation.

The formula for the differentiable distance transform is:

$$D(\alpha_i^l) = -\lambda \log(\alpha_i^l * \exp\left(-\frac{d(\cdot, 0)}{\lambda}\right)). \tag{4.9}$$

This transformation uses convolution $*$ with a kernel that smooths the distance field, making it differentiable.

```
@jit
def CDT(I, k):
```

```
    l = 0.35
    n,h,w,c = I.shape
    [X,Y] = jax.numpy.meshgrid(jax.numpy.linspace(-jax.numpy.floor(k/2),jax.numpy.
                                                  floor(k/2),num=7), jax.numpy.linspace(
                                                  -jax.numpy.floor(k/2),jax.numpy.floor(
                                                  k/2),num=7))
    dis_0 = jax.numpy.exp(-jax.numpy.sqrt(X**2 + Y**2)/l)
    dis = jax.numpy.reshape(jax.numpy.dstack([dis_0]*n),(n,c,7,7))
    return -l * jax.numpy.log(lax.conv_general_dilated(I,dis,window_strides=(1,1),
                                                  padding='SAME',dimension_numbers=('
                                                  NHWC', 'OIHW', 'NHWC')))

@jit
def CCDT_2(I, s, k):

    n,h,w,c = I.shape

    D_star = jax.numpy.reshape(jax.numpy.nan_to_num(CDT(I,k),posinf=0.0)[:,:,:,0],(n
                                                  ,h,w,c))

    flat = D_star > 0
    D = D_star
    I = I + flat
    pad = jax.numpy.floor(k/2)
    def for_loop(idx, input):
        conv = CDT(input[1],7)
        D_star = jax.numpy.reshape(jax.numpy.clip(jax.numpy.nan_to_num(conv ,posinf=
                                                      0.0),0)[:,:,:,0],(n,h,w,c))
        flat = D_star > 0
        I = input[1] + flat
        D = input[0] +  (idx * pad) * flat +  D_star
        return D, I

    D, I = lax.fori_loop(lower=0, upper=50, body_fun=for_loop, init_val=(D,I))
    return D
```

Moreover, to facilitate differentiability in the Chan–Vese optimisation problem, as is conventionally done, we employ smooth approximations of the Heaviside and Dirac delta functions.

Figure 4.11: This diagram demonstrates the incorporation of the Chan-Vese method within the attention mechanism. The resized input image undergoes a distance transform, and the initial contour is formed for the Chan-Vese algorithm to segment the image iteratively, refining the attention focus on key regions for segmentation tasks. The symbols $\bigoplus$ and $\bigotimes$ represent the addition and multiplication of the tensors.

4.3 Experiments

In this study, we used the TCGA LGG database, an openly available online repository [Ped] containing magnetic resonance imaging (MRI) 2D images of brain tumour patients. The database comprises 110 patients from The Cancer Genome Atlas (TCGA) lower-grade glioma collection, with genomic cluster data and at least one fluid-attenuated inversion recovery (FLAIR) sequence available. Table 5.1 summarises the experimental results. We have used 2D MRI images as our learning and training datasets. We have set aside ten patients' data to form an as-independent test set as possible (whereas two images from the same patient can be separated in the training and validation set, test data are always the result of a different acquisition from the training and validation set).

Implementation Details We used a large batch of 32 for gradient update, and the model parameters are optimised using an adamW optimiser [Los17] with learning rate 5×10^{-4} and batch normalisation. We applied standard data augmentation (resize, horizontal flip, vertical flip, random rotate, transpose, shift and scale, normalise). The Chan-Vese parameters μ and ν are set respectively to 0.1 and 1.0. To optimise the model's performance and to penalise the error between the prediction mask x and the ground truth mask y, we used a combination of Dice loss and Binary Cross-Entropy (BCE) loss (Equation 5.8).

$$\mathcal{L}_S(x, y) = \frac{2 \times \sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i + \sum_{i=1}^{N} y_i} + \frac{1}{N} \sum_{i=1}^{N} -(y_i \log(x_i) + (1 - y_i) \log(1 - x_i)). \tag{4.10}$$

The Dice loss helps manage class imbalance, while BCE loss ensures stable learning, making the combination of these losses effective for accurate mask prediction. Specifically, Dice loss emphasises overlap between the prediction and ground truth (See Figure 4.12), particularly for smaller regions, while BCE provides pixel-wise precision and helps optimise the probability outputs during training. The added attention layer slows the training by an average of 1 sec out of 7 sec per batch. The code is written in Jax using the Haiku framework. The implementation parameters are summarised in Figure 4.13.

4.3.1 Segmentation Results

This study compares our proposed attention-based U-Net model with the classical U-Net and the original Attention U-Net. The experimental results are summarised in Table 5.1. Our proposed model demonstrated superior performance in terms of Intersection over Union (IOU)

Table 4.1: Segmentation results (IOU) on the TGCA_LGG brain MRI database. Significant results are highlighted in bold font

Name	Dice	IOU	Hausdorff	FPR	FNR
UNet	0.832	0.829	2.390	0.010	0.013
Attention UNet	**0.830**	0.833	2.416	**0.009**	0.015
Chan-Vese UNet	0.824	**0.848**	**2.329**	0.012	**0.013**

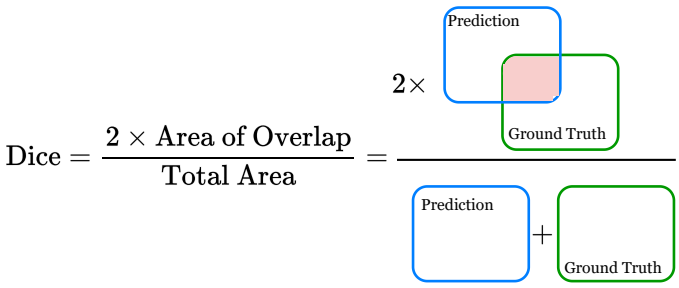


Figure 4.12: Illustration of predicted segmentation versus ground truth segmentation for evaluating the Dice metric.

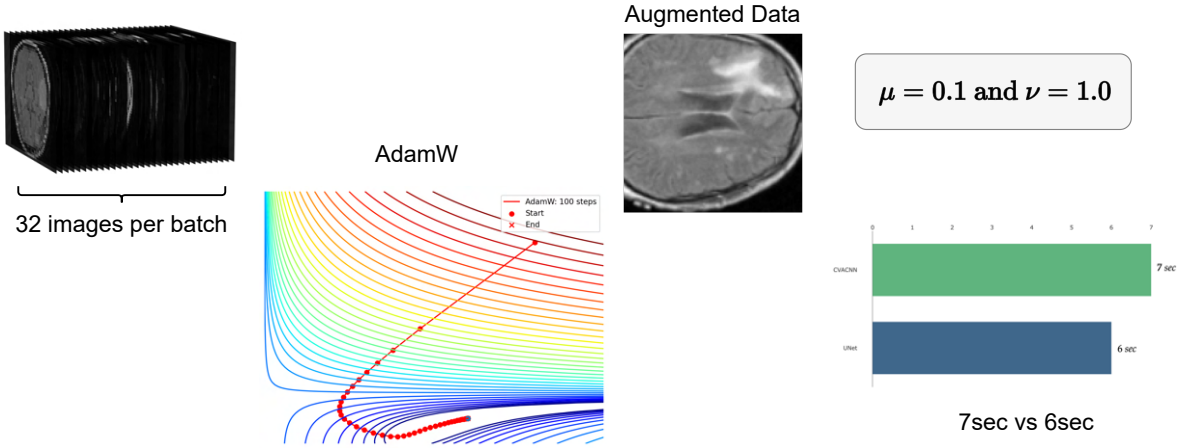


Figure 4.13: This diagram showcases the key hyperparameters used in the model, such as batch size (32 images per batch), optimisation algorithm (AdamW), and the impact of augmented data on training speed (7 seconds per batch vs. 6 seconds per batch)

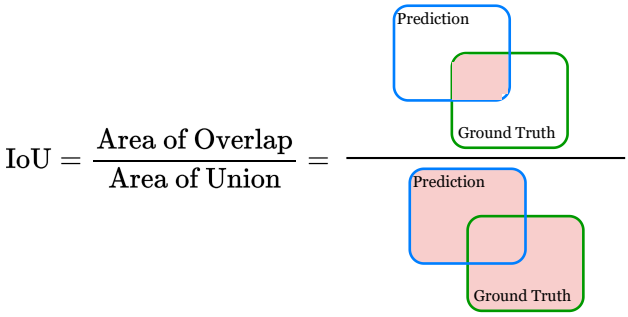


Figure 4.14: Illustration of predicted segmentation versus ground truth segmentation for evaluating the Intersection over Union (IoU) metric. The overlap and differences between the two masks are used to compute the IoU score.

scores and improved false negative performance.

The IOU score is a commonly used metric for evaluating the accuracy of image segmentation. It measures the overlap between the predicted and ground truth segmentation (See Figure 4.14). A higher IOU score indicates better segmentation accuracy. Our proposed model achieved higher IOU scores than the classical U-Net and the original Attention U-Net, suggesting that it produces more accurate segmentation results.

In addition to the improved IOU scores, our proposed model demonstrated improved false negative performance. False negatives occur when the model fails to detect a region of interest in the input image. This can be particularly problematic in medical image segmentation, where missing a tumour or other abnormality can have serious consequences. Our proposed model's ability to focus on a smaller area of interest and integrate the Chan-Vese method enables more effective capture of relevant information, reducing the risk of information loss and improving false negative performance.

The integration of the Chan-Vese method in our proposed model provides several benefits. First, it allows the network to perform intermediate segmentation for each block of the expansion phase, improving the final segmentation's accuracy. Second, it provides a control signal to facilitate learning and enhance the network's convergence. Finally, it allows the network to focus on the tumorous region of the input image, which is often the most critical region for medical image segmentation.

In summary, our proposed attention-based U-Net model demonstrated superior performance regarding IOU scores and false negative performance compared to the classical U-Net and the original Attention U-Net. This can be attributed to the model's ability to focus on a smaller area of interest and the integration of the Chan-Vese method, which enables more effective capture of relevant information and reduces the risk of information loss. These results demonstrate the potential of our proposed model for improving the accuracy of medical image

segmentation.

4.3.2 Chan-Vese Attention Masks analysis

The results of the attention layer, as shown in Figure 4.15, demonstrate the effectiveness of our proposed attention-based U-Net model with the integration of the Chan-Vese method. With Chan-Vese, the attention mask quickly converges to a tumour-like segmentation. This is achieved by taking advantage of the minimisation of the Chan-Vese energy from the initialisation of the mask, which is inspired by the attention method of Oktay et al. [Okt18]. Additionally, using the initial image to be segmented further enhances the accuracy of the segmentation.

As the segmentation process progresses, the contours of the tumour become more precise, and the active intensity of the tumour becomes the confidence in the energy to be minimised. This is achieved using the 0 level set, which enables the neural network to selectively prioritise the tumour area during segmentation. By focusing on the tumour area, the network can better capture the relevant features and reduce the risk of information loss.

Once the tumour area has been prioritised, the upper-level set is subsequently employed to refine the segmentation. This step further improves the accuracy of the segmentation by refining the boundaries of the tumour and ensuring that the segmentation closely matches the ground truth.

Overall, the results of the attention layer demonstrate the effectiveness of our proposed attention-based U-Net model with the integration of the Chan-Vese method for medical image segmentation. By quickly converging to a tumour-like segmentation and prioritising the tumour area during segmentation, the network can better capture the relevant features and reduce the risk of information loss. The subsequent use of the upper-level set further improves the accuracy of the segmentation, resulting in more precise and accurate segmentation results.

4.3.3 Comparison with Attention UNet

Figure 4.16 compares the attention output of our proposed Chan-Vese Attention Module and the classical Attention Module. Both methods allow the neural network to focus on the tumour area, highlighting the importance of attention mechanisms in medical image segmentation.

However, it should be noted that the method proposed by Oktay et al. [Okt18] obtains a finer mask on specific details of the tumour but needs to manage to rank the confidence of the presence of the tumour in the framework of our study. This can result in artefacts outside the tumour area that do not correspond to the object of interest in the image. These artefacts can be problematic as they can lead to false positives and reduce the accuracy of the segmentation.

In contrast to these observations, our proposed Chan-Vese Attention Module focuses only on the tumour area inside the skull. This is achieved by integrating the Chan-Vese method into the attention mechanism, enabling the network to capture the relevant features better and reduce the risk of information loss. By focusing only on the tumour area, the network can produce more accurate segmentation results and reduce the risk of false positives.

Furthermore, the Chan-Vese method in the attention mechanism provides several benefits. It allows the network to perform intermediate segmentation for each block of the expansion phase, improving the final segmentation's accuracy. It also provides a control signal to facilitate

learning and enhance the network's convergence. Finally, it allows the network to focus on the tumorous region of the input image, which is often the most crucial region for medical image segmentation. The last image in Figure 4.17 shows the confidence of the segmentation as level-set.

Figure 4.15: Visualisation of attention masks overlaid on MRI brain scans highlighting tumour regions. The first column shows the original MRI images, while the subsequent columns represent attention maps generated by the model across different learning iterations (learning iteration: 1, 50, 100, 200, 300) from left to right. The attention focuses progressively on the tumour (highlighted in red), with surrounding areas depicted in varying colour intensities to illustrate the model's focus on significant regions.

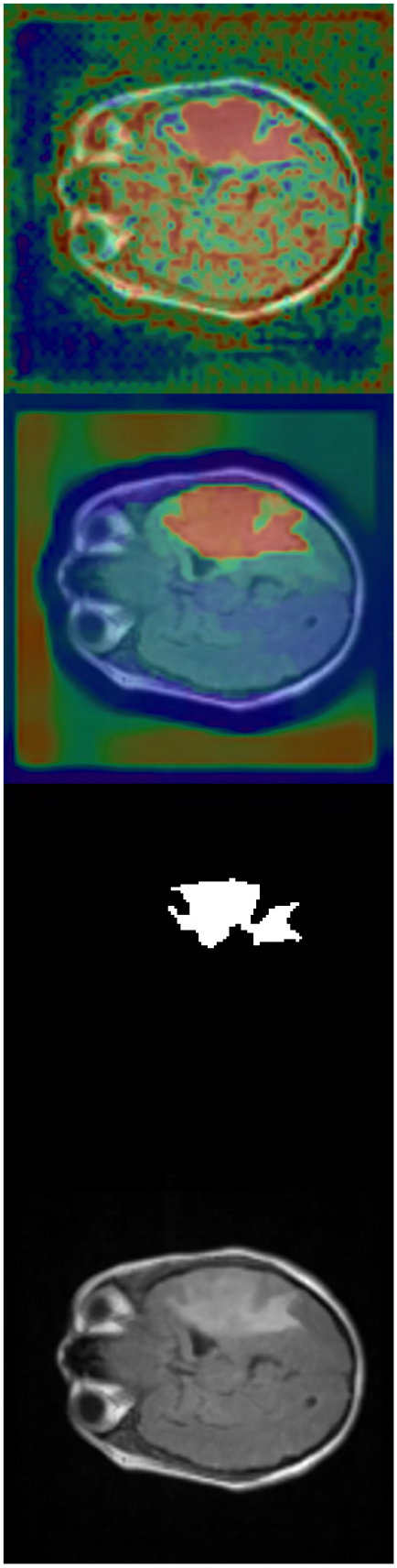


Figure 4.16: Comparison of the Attention Mask between a Chan-Vese Attention and the Original Attention. From left to right: the input MRI scan, the tumour segmentation mask (ground truth), the Chan-Vese attention mask highlighting the segmented region, and the original attention mask. The differences between the Chan-Vese and original attention methods illustrate how each approach focuses on the tumour area, with varying degrees of sensitivity and noise in the surrounding regions.

4.4 Partial Conclusion

This chapter presents a novel approach to image segmentation that combines classical energy minimisation techniques with deep learning architectures. Specifically, we introduced the *Chan-Vese Attention Gate*, an attention mechanism integrated into the U-Net model to enhance segmentation accuracy. By incorporating the Chan-Vese energy minimisation within the attention gates, our method allows for more precise control over the segmentation masks and enables the gradient to propagate through the optimisation process.

Our approach addresses the critical challenge of robust object segmentation, particularly in medical imaging applications where precise delineation is essential. Traditional convolutional neural networks like U-Net have demonstrated impressive performance in segmentation tasks but often cannot explicitly integrate geometric and topological constraints. By embedding the Chan-Vese model into the attention mechanism, we leverage prior knowledge about the object's shape and structure, leading to improved segmentation results.

Through comprehensive experiments on the TCGA LGG brain MRI database, we demonstrated that our method achieves competitive results in binary segmentation tasks. The integration of the Chan-Vese Attention Gate improved the Intersection over Union (IoU) scores compared to the classical U-Net and the original Attention U-Net and reduced false negatives, which is crucial in medical diagnostics. The attention masks generated by our model focused on the tumour regions, confirming the effectiveness of our approach in capturing spatial information relevant to the regions of interest.

Moreover, our method maintains computational efficiency, with only a slight increase in computation time due to the added attention mechanism. The differentiability of the optimisation problem was addressed by employing a differentiable variant of the Euclidean distance transform, ensuring seamless integration into the neural network's training process.

In conclusion, the Chan-Vese Attention Gate offers a promising direction for enhancing segmentation models by combining the strengths of classical energy minimisation and deep learning. This hybrid approach provides more control over the segmentation process and can be particularly beneficial in medical imaging, where accuracy and reliability are very important. Future work could extend this mechanism to multi-class segmentation problems and investigate its applicability to other imaging modalities and domains. Additionally, integrating other geometric constraints and further optimising computational efficiency could broaden the impact and utility of this method.

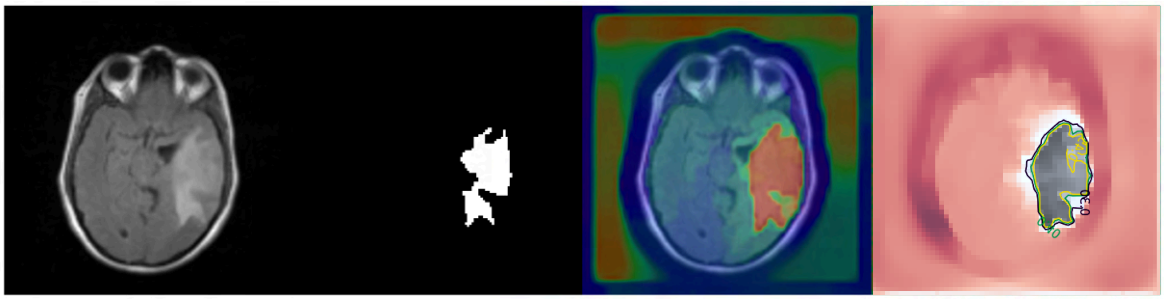


Figure 4.17: Level set recovered by Chan-Vese alongside the energy surfaces

Chapter 5

Fast Marching Energy CNN

Objectifs

In this chapter, we focus on solving a segmentation problem by leveraging geodesic distances and the rich geometrical information they provide, which is essential for many imaging applications. Geodesic distance computation has been used for image segmentation for a long time using Image-based metrics. We introduce a new method by generating isotropic Riemannian metrics adapted to a problem using CNN and illustrate an application example. We then apply this idea to the segmentation of brain tumours as unit balls for the geodesic distance computed with the metric potential output by a CNN, thus imposing geometrical and topological constraints on the output mask. We show that geodesic distance modules work well in machine learning frameworks and can achieve state-of-the-art performances while ensuring geometrical and topological properties.

Contents

5.1 Introduction

This is a joint work with Théo Bertrand and Laurent D. Cohen. It was accepted at the SSVM 2023 conference and published online as part of the Proceedings of the 9th International Conference on Scale Space and Variational Methods in Computer Vision (SSVM 2023).

In this chapter, we aim to segment a region defined as the set of points in the image whose geodesic distance from a source point falls below a specified threshold. In this context, we consider the region to be a geodesic ball with respect to the learned metric.

Geodesic curves and distances have been extensively used to convey geometric properties in various applications, from computer vision and medical imaging to robotics [Rat09] and geophysics. Traditional methods in these domains typically rely on prior knowledge of the task to explicitly construct a Riemannian metric g from data. This metric is then used to compute geodesic distances, which are integral to understanding the underlying geometry of the data. However, this approach introduces a bias in the choice of the metric tensor, as it requires users to make arbitrary decisions and perform manual parameter tuning.

To address this issue, the approach presented in this work aims to eliminate the bias introduced by manual metric selection. We propose a novel framework that generates the metric tensor directly from data using a Neural Network architecture. The parameters of this neural network are optimised through a supervised learning approach using training data. By learning the metric from data, we circumvent the need for subjective decisions and fine-tuning, making the process more objective and adaptable. The neural network architecture employed in this work is designed to take raw data as input and output a metric tensor that best represents the geometric structure of the data. The network is trained using a loss function that measures the discrepancy between two segmentation masks, one generated using the image's predicted metric and the geodesic distance computation. By minimising this loss, the network learns to create a metric that accurately captures the intrinsic geometry of the data.

The use of geodesic distances in segmentation tasks has a long history. To the authors' knowledge, the first article to segment an image's region using a minimal path distance and fast marching is Malladi et al. [Mal98], with application on a 3D brain image. For the segmentation of tubular tasks, we can refer to Chen et al. [Che16] for instance, a method that segments the 3D vascular tree by propagating the front of the minimal path distance computation. Similarly, Cohen et al. [Coh07] segments vascular structures by introducing an anisotropic metric, determined dynamically by evaluating local orientation scores during the Fast Marching computations. Those three articles already use the level sets of the geodesic distance (or "geodesic balls") to provide the segmentation mask. We also mention Cohen et al. [Coh97]. They present a boundary detection method that finds the global minimum of an active contour model's energy, improves initialisation, avoids local minima, and detects closed contours by minimising path length in a Riemannian metric using an efficient numerical method. These works generally avoid treating the task holistically and focus on providing a good model for segmenting structures. In contrast, this work tries to treat the problem end-to-end and generalise to a large

dataset of input images.

Only a few previous methods are interested in learning a metric from data. We may mention recent works such as Scarvelis et al. [Sca22] and Heitz et al. [Hei21] that try to find metric tensors that fit spatio-temporal data to capture the velocity fields and underlying geometry of the data. The first paper models trajectories as the solutions of a dynamic system generated by a neural network, considering the dynamics of the whole population by penalising an optimal transport cost between two consecutive timestamps. However, [Hei21] tries to interpolate a sequence of histograms with Wasserstein barycenters by optimising over the metric tensor appearing in the ground cost. Also, there are important links between the Wasserstein optimal transport, its dynamical formulation and geodesics; for further reading, we refer to Ambrosio et al. [Amb21]. These works propose interesting frameworks but must be more focused on generalising the generation of the metric tensors.

[Ben10] is an older article important for our work, as they laid the ground for differentiating the geodesic distance concerning the metric in the Fast Marching algorithm. They then proceed to apply it in the setting of inverse problems to retrieve the metric from distance measurements. Its only concern was to solve inverse problems involving the geodesic distance, whereas we go one step further by including a Fast Marching module in a deep learning segmentation procedure. The sub-gradient marching algorithm is briefly described in section 2 as it is essential to our framework to propagate through the Fast Marching module and carry the learning step.

Regarding Deep Learning, please refer to Chapter 3. For a review of deep learning methods in medical imaging, one might refer to Zhou et al. [Zho21]. The general techniques of directly producing segmentation from medical images are already quite efficient. Still, they need more robustness and impose more structure on the segmentation that comes out of the network. Contrary to this, our work imposes many constraints on the topology of the segmented region (namely, a set with trivial topology).

The rest of the chapter is structured as follows. We present the isotropic problem, where the metric tensor is uniform in all directions. Following this, we delve into the fast marching method, a popular algorithm for computing geodesic distances efficiently. We explain how geodesic distances and their gradients can be calculated using sub-gradient methods, providing the mathematical foundation for understanding how geodesic distances are computed and how they can be used to analyse data geometry.

Next, we introduce our method for Fast Marching Energy CNN. This novel architecture combines the strengths of convolutional neural networks and fast marching methods for efficient and accurate geodesic distance computation. This model leverages the learned metric tensor to compute geodesic distances that respect the underlying geometric structures of the data. We then present the main results of our experiments on the brain tumour MRI image dataset. We compare the performance of our method with traditional approaches that rely on manually designed metrics. The results show that our method achieves higher segmentation accuracy and a remarkable ability to learn from data and generalise to unseen data.

Subsequently, we discuss the limitations of the isotropic approach and introduce the anisotropic problem, where the metric tensor varies with direction. This section highlights the need for methods to handle directional information and the challenges associated with anisotropic metrics. We explore an alternative approach to computing geodesic distances using the heat equa-

tion and Varadhan's formula. This method leverages automatic differentiation to calculate the gradients of geodesic distances, providing a more flexible and efficient way to analyse the geometry of data.

We then present a modified version of our model that uses an approach similar to the isotropic model but incorporates directional information. This model aims to address the limitations of the isotropic approach while maintaining the efficiency of the fast marching method. Following this, we introduce a second model that uses a probability map and Kullback-Leibler (KL) regularisation to handle the anisotropic problem. This model leverages probabilistic methods to capture the directional information and regularise the metric tensor, leading to more accurate geodesic distance computations.

We present the results of our experiments using the modified models on the brain tumour MRI image dataset. We compare the visual performance of Model 1 and Model 2 with the original isotropic model and discuss the improvements achieved by incorporating directional information. Finally, we conclude with a summary of our contributions and potential directions for future research. We discuss the implications of our work for medical imaging and other applications and highlight the potential of learning-based approaches for geometric data analysis. By structuring the chapter in this manner, we aim to provide a comprehensive overview of our approach to learning geodesic distances from data, addressing both the isotropic and anisotropic problems and demonstrating the effectiveness of our methods through extensive experiments.

5.2 Isotropic Geodesic Case

5.2.1 Computing geodesic distances and their gradient

The geodesic distance is a fundamental concept in the field of Riemannian geometry (See Section 2.4), and it is used to quantify the distance between two points on a (compact, path-connected) manifold $\mathcal{M}$. It is defined as the minimal length of all possible paths linking two points on the manifold.

Formally, the geodesic distance is given by the following :

$$d_g(x,y) = \inf_{\gamma \in \mathrm{Lip}([0,1],\mathcal{M}),\gamma(0)=x,\gamma(1)=y} \int_0^1 \sqrt{g_{\gamma(t)}(\gamma'(t),\gamma'(t))}\,\mathrm{d}t, \tag{5.1}$$

where $\mathrm{Lip}([0,1],\mathcal{M})$ is the space of Lipschitz curves on the manifold $\mathcal{M}$ and parameterized by the interval $[0,1]$. g is a metric tensor, which is a map defined at each point $x \in \mathcal{M}$ as $g_x : (u,v) \in \mathcal{T}_x\mathcal{M}^2 \mapsto g_x(u,v)$ is positive definite bilinear form. This means that $\sqrt{g_x}$ is a Euclidean norm on $\mathcal{T}_x\mathcal{M}$, the tangent space to $\mathcal{M}$ at point x.

In this work, we will consider a straightforward mathematical framework, where $\mathcal{M}$ is simply a path-connected, open and bounded set Ω of $\mathbb{R}^d$ and $\mathcal{T}_x\mathcal{M}$ can be identified with $\mathbb{R}^d$. This simplification allows for a more straightforward implementation of the geodesic distance while maintaining its core properties and mathematical foundation. In this section, we will have $g_x(u,v) = \phi(x)^2 \langle u,v\rangle_{\mathbb{R}^d}$.

5.2.2 Recall on the Fast Marching Algorithm

Since the seminal work of Sethian [Set96], the Fast Marching algorithm has been one of the most widely used methods for computing geodesic distances on a manifold. The Fast Marching method computes the geodesic distance by front propagation.

The Eikonal equation (See Section 2.4.1) has the geodesic distance as its unique positive viscosity solution and is critical to front propagation in Fast Marching.

The distance u from a set $S \subset \Omega$ satisfies the Eikonal equation:

$$\begin{cases} \forall x \in \Omega \setminus S, & \|\nabla u(x)\| = \phi(x), \\ \forall x \in S, & u(x) = 0, \end{cases} \tag{5.2}$$

It can be shown that the unique positive solution to the equation 5.2 in the sense of viscosity solutions is the geodesic distance from the set S, relative to the metric tensor field associated with the matrices $\phi(x)^2 \mathbf{I}_d$.

The Eikonal equation in dimension 2 is discretised using the upwind scheme introduced by Cohen et al. [Coh97]:

$$\sum_{1 \leq i \leq 2} \frac{1}{h^2} \max(u_p - u_{p+e_i}, u_p - u_{p-e_i}, 0)^2 = \phi_p^2, \tag{5.3}$$

with u_p and ϕ_p the geodesic distance and potential at point p in the discretized domain Ω, $p \pm e_i$ denote the adjacent points on the grid and h is the discretization parameter. This algorithm allows us to compute distances with less bias due to the grid approximation of space.

Fast Marching (See Section 2.4.2) is an algorithm that iteratively visits each point on the grid from neighbour to neighbour. At each iteration, we look at the neighbour points to those already accepted and take the nearest point among the neighbours. We repeat this by computing the new neighbourhood of the accepted points. We initialise all values at $+\infty$ except the seed point at 0. Depending on the number of accepted points connected to p on the grid, equation (5.3) reduces either to a quadratic of the affine equation to find u_p from the values of the parent points.

In practice, we use the python library *Hamiltonian Fast Marching* (*HFM*) [Mir19] that provides a fast and efficient implementation of the Fast Marching method and the so-called Subgradient Marching Algorithm.

5.2.3 Differentiating Fast Marching

Differentiating the geodesic distance with respect to the metric is an essential tool in many applications, such as shape optimisation and optimal control. The first work to propose a numerical method to differentiate the geodesic distance with respect to the metric is by Benmansour et al. [Ben10], and it has found many applications (see, for instance, [Bon20]).

To compute the derivatives of the Fast Marching algorithm, we rely on the subgradient marching algorithm. The goal at hand is to optimise the geodesic distance with regard to the metric ϕ. To do so, we derive a subgradient for the metric ϕ by perturbating the metric:

$$\frac{\mathrm{d}d_{\phi+\varepsilon\xi}(x,y)}{d\varepsilon}\Big|_{\varepsilon=0} = \int_{\gamma^\star} \xi, \tag{5.4}$$

where $\gamma^\star$ is a minimiser for the ϕ-length functional. This formulation is hard to differentiate in a discretised robust manner. This is why Benmansour et al. [Ben10] approach proposed to discretise the approximation of the distance by the Fast Marching algorithm rather than the direct geodesic distance.

We can use the discretised Eikonal equation (5.5) update to differentiate the geodesic distance. By taking the Eikonal equation written in dimension 2 and using the setting of interest, i.e. an isotropic metric $g_x(v,w) = \phi(x)^2 \langle v, w\rangle_{\mathbb{R}^2}$, we write the discretised version of the Eikonal equation, with h the discretisation parameter, the discretised domain is simply a regular square grid :

$$\begin{cases} (u_p - u_{p\pm e_1})^2 + (u_p - u_{p\pm e_2})^2 = h^2\phi_p^2 \text{ if p has 2 parents,} \\ u_p = \min_i u_{p\pm e_i} + h\phi_p \text{ if p has only 1 parent or } h^2\phi_p^2 < (u_{p\pm e_1} - u_{p\pm e_2})^2. \end{cases} \tag{5.5}$$

Thus u_p is the value of the distance computed by fast marching at point p, and we define $D_\phi u_p \in \mathbb{R}^{n^2}$ the differential of u_p with respect to the potential ϕ.

Differentiating with respect to ϕ in the two cases of update, we get

$$\begin{cases} (u_p - u_{p\pm e_1})(D_\phi u_p - D_\phi u_{p\pm e_1}) + (u_p - u_{p\pm e_2})(D_\phi u_p - D_\phi u_{p\pm e_2}) = h^2\phi_p \\ \text{if p has 2 parents,} \\ D_\phi u_p = D_\phi u_{p\pm e_i} + h\mathbb{1}_p \text{ if p has only 1 parent or } h^2\phi_p^2 < (u_{p\pm e_1} - u_{p\pm e_2})^2, \end{cases} \tag{5.6}$$

with $\mathbb{1}_p \in \mathbb{R}^{n^2}$ the vector filled with zero except at coordinate p, which gives the update:

$$\begin{cases} D_\phi u_p = \frac{(u_p - u_{p\pm e_1})D_\phi u_{p\pm he_1} + (u_p - u_{p\pm e_2})D_\phi u_{p\pm e_2} + h^2\phi_p}{(u_p - u_{p\pm e_1}) + (u_p - u_{p\pm e_2})} \text{ if p has 2 parents,} \\ D_\phi u_p = D_\phi u_{p\pm e_i} + h\mathbb{1}_p \text{ if p has only 1 parent or } h^2\phi_p^2 < (u_{p\pm e_1} - u_{p\pm h_2})^2, \end{cases} \tag{5.7}$$

This update can then be used to compute the gradient of the geodesic distance with respect to the metric tensor during the Fast Marching iterations. This method introduced in Benmansour et al. [Ben10] is named *Subgradient Marching Algorithm*. This method can be extended to higher dimensions and more general Finsler metrics.

5.2.4 Model

The proposed method presented in this study uses a neural network, specifically a modified version of the U-Net architecture, to segment regions of an image as geodesic balls with respect to a metric. The metric is obtained by training a convolutional neural network (CNN) to provide both the metric and the centre or seed of the geodesic ball. The framework, as shown in Figure 5.1, processes the input image using the encoder component of the U-Net, resulting in a vector representation of the image. This vector is then passed through two separate decoders to perform distinct tasks.

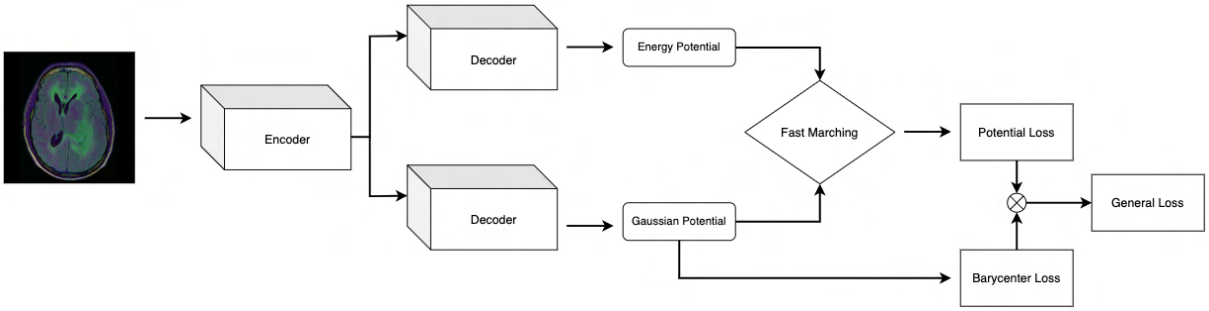


Figure 5.1: Diagram of the framework from the input image to the loss.

The first decoder predicts the potential ϕ to be used by the fast marching module, which can be computed using the HFM library. The second decoder predicts a Gaussian potential that represents the probability of the presence of the region's barycenter in a given area, which is also provided as a seed to the fast marching module. The distance map generated by the fast marching procedure is then used to find a geodesic ball for segmentation. The expected segmentation is compared to the predicted segmentation, and the theoretical barycenter is compared to the predicted Gaussian potential to compute the error.

The distance computation module can be written as a function of seed points and input metrics. The metric ϕ is defined as the output of a CNN architecture, such as the widely used U-Net, with θ being in the space of parameters. We enforce positive and non-zero properties of the metric by taking $\phi = f_\theta(u)^2 + \epsilon$, with u being the input image and let f_θ be a CNN, with $\theta \in \mathbb{R}^p$ the space of parameters. To avoid solutions that distribute a lot of mass everywhere, as noted in [Ben10], we ensure that the total mass of the metric is reasonable by applying a transformation $\phi \mapsto \frac{\phi}{\max(\frac{1}{\lambda}\|\phi\|_1, 1)}$ that upper bounds the L^1 norm at a fixed level λ (We chose in this work to empirically bound the total mass at 5).

In this study, we focus on the potential generation and employ two different architectures commonly used for image segmentation: the U-Net introduced by Ronneberger et al. [Ron15] and a combination of the U-Net and ResNet ([He16]). The U-Net is a fully convolutional neural network designed for image segmentation, comprising a contracting and expansive path. The contracting path reduces the spatial resolution of feature maps, while the expansive path increases it. Combining these paths allows for extracting high-level features from the input image and recovering the spatial resolution to provide a segmented output.

However, CNNs' depth can cause vanishing gradients, affecting model performance. To address this, we propose using ResNet-U-Net, a combination of the U-Net and ResNet-34 model, in the encoder portion of the network. ResNet-34 benefits from deep residual learning and comprises a 7x7 convolutional layer, a max pooling layer, and 16 residual blocks.

By combining these architectures, ResNet-U-Net can capture fine and coarse features of input images and learn deeper and more complex representations. This results in a more accurate and robust model for image segmentation tasks, as demonstrated by our experimental results. Additionally, we introduced modifications to the expansive path of both networks, implementing a dual expansive path system to predict potential energy and a Gaussian potential for predicting the barycenter. These modifications are illustrated in Figure 5.1. Overall, our

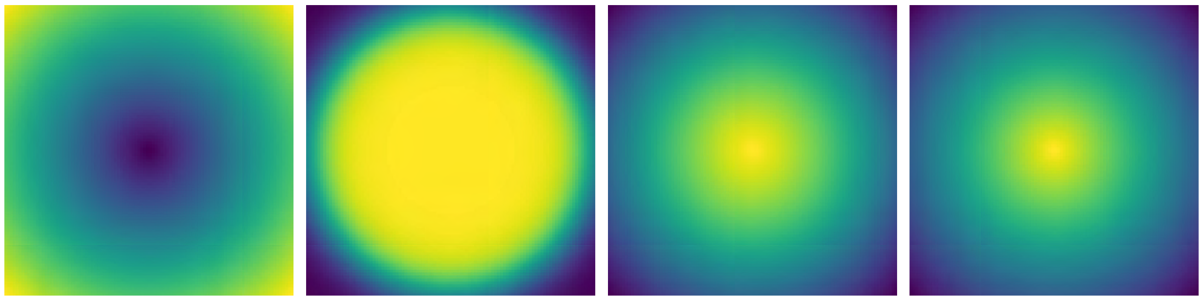


Figure 5.2: The image compares the original distance map centred at the source point $x_0 = (0, 0)$ with sigmoid-based masks for different values of δ. As δ decreases, the sigmoid approximation more closely resembles the characteristic function of the unit ball, with sharper transitions occurring for smaller δ values and smoother transitions for larger δ values. (From second to last image: $\delta = 0.1, 0.5,$ and 1)

proposed model demonstrates promising results for potential generation tasks.

5.2.5 Generating masks with geodesic balls

Applications may take advantage of topological priors on the label to reconstruct. For instance, one may need to recover regions in an image we know to be path-connected and of trivial topology. Such regions might be modelled as balls related to a specific distance and recovered as indicator functions of such a ball. Formally, we expect for a set E to recover a characteristic function as $\chi_{d_\phi(x_0,\cdot)\leq 1}$ for well chosen $x_0 \in \mathbb{R}^d$ and $\phi \in L^1(\Omega)$.

With this method of building masks for specific tasks, we can generalise using a neural network architecture and find good potential ϕ to segment attractive regions in images. To do this, we would need to compute the gradient of a chosen loss function and thus would need to differentiate the mask; that is why we will replace the indicator function on the unit ball, which would yield zero gradients almost everywhere, by a sigmoid that will smoothly interpolate between the value 1 in the region inside the unit ball and 0 outside. Given the distance map $d_\phi(x_0, \cdot)$, our mask then becomes $\chi^\delta(d_\phi(x_0, \cdot)) = 1 - \frac{1}{1+\exp(-(d_\phi(x_0,\cdot)-1)/\delta)}$, which approaches characteristic function of the unit ball as the parameter δ approaches 0 (See Figure 5.2). δ will typically be taken off the order of the pixel size, i.e. approximately the inverse of the image size.

Figure 5.3 shows how it is possible to approach the characteristic function of different sets with this formulation. This problem is not convex so that solutions may vary depending on the initialisation. Most of the time, potentials converge to a solution that puts a lot of mass on the edges of the mask to recover. The seed here is fixed to x_0, the centre of the balls to be fitted, and the potential ϕ is directly optimised using automatic differentiation and ADAM with a "learning rate" equal to 0.01. ϕ^2 is taken as input for the fast marching algorithm instead of ϕ as an easy way to enforce the positivity of the potential smoothly.

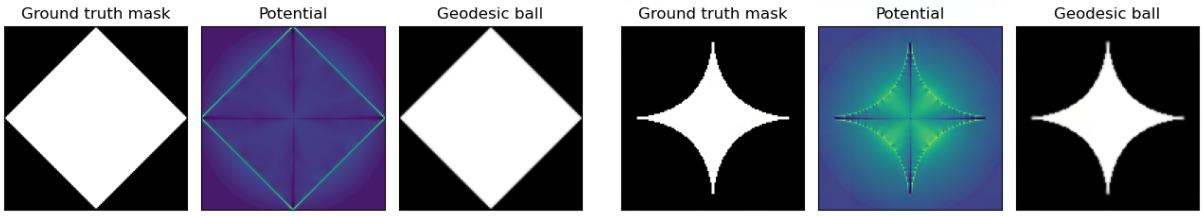


Figure 5.3: Example of recovery of an isotropic metric fitting two regions by minimising $\|\chi^{\delta} \circ d_{\phi^2} - y\|_2^2$ with respect to ϕ, where y is the ground truth mask, $\delta = 0.01$. x_0 is taken as the centre of the mask to be recovered.

5.3 Experiments

As announced before, our experiments were led on a tumour segmentation task.

5.3.1 Data

The use of isotropic fast marching is particularly well-suited for the type of data we are working with, specifically the Brain MRI segmentation tasks from the TCGA LGG database [Ped]. This is the same dataset used in Section 4.3. We recall that this dataset contains MRI scans of 110 patients with brain tumours, resulting in 1189 images. The scans include fluid-attenuated inversion recovery (FLAIR) sequences and genomic cluster data. Isotropic fast marching ensures that the propagation of the front is uniform in all directions. Uniform propagation helps accurately capture the tumour boundaries regardless of their orientation. MRI images, especially those involving tumours, can be noisy due to factors such as patient movement or imaging artefacts. Isotropic fast marching helps reduce the effect of the noise, ensuring that the segmentation process is not significantly affected by minor irregularities in the image data.

We have used 2D MRI images as our learning and training datasets. We have set aside ten patients' data to form an as-independent test set as possible (whereas two images from the same patient can be separated in the training and validation set, test data are always the result of a different acquisition from the training and validation set).

We applied data augmentation on the training images to increase the training set's diversity and improve the model's generalisation. The data augmentation techniques used were horizontal flipping with probability $p = 0.5$, vertical flipping with probability $p = 0.5$, random 90-degree rotation with probability $p = 0.5$, transpose with probability p=0.5, and a combination of shifting, scaling, and rotating with probability $p = 0.25$. We set the shift, scale, and rotation limits to 0.01, 0.04, and 0 (as we already perform rotation). We computed the tumour seed using an Euclidean barycenter of the mask region.

5.3.2 Model Training Procedures

In this study, U-Net architecture was employed for image segmentation. The model was initialised with Kaiming initialisation [He15]. It is an initialisation method for neural networks that considers the non-linearity of activation functions, such as ReLU activations. We optimised the network weights using the Adam optimiser [Kin14], which has been widely used

in literature due to its capability to adjust the learning rate during training. The learning rate was set to 1e-3, a commonly used value in CNNs, as it provides a balance between achieving convergence and avoiding overshooting the optimal solution. To optimise the model's performance and to penalise the error between the prediction mask x and the ground truth mask y, we used a combination of Dice loss and Binary Cross-Entropy (BCE) loss (Equation (5.8)).

$$\mathcal{L}_S(x, y) = \frac{2 \times \sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i + \sum_{i=1}^{N} y_i} + \frac{1}{N} \sum_{i=1}^{N} -(y_i \log(x_i) + (1 - y_i) \log(1 - x_i)). \tag{5.8}$$

The Dice loss helps manage class imbalance, while BCE loss ensures stable learning, making the combination of these losses effective for accurate mask prediction. Specifically, dice loss emphasises the overlap between prediction and ground truth, particularly for smaller regions, while BCE provides pixel-wise precision and helps optimise probability outputs during training.

A binary cross-entropy loss was used to control the error in the seed prediction where h^1 is the predicted seed, and h^2 is the ground truth seed.

$$\mathcal{L}_H(h^1, h^2) = \frac{1}{N} \sum_{i=1}^{N} -(h_i^2 \log(h_i^1) + (1 - h_i^2) \log(1 - h_i^1)) \tag{5.9}$$

The final loss is:

$$\mathcal{L}(x, y, h^1, h^2) = \mathcal{L}_S(x, y) + \mathcal{L}_H(h^1, h^2) \tag{5.10}$$

The Dice loss function, known for handling imbalanced data, was combined with the BCE loss function, which provides stability during training.

To determine the distance between two barycenters, a transformation of the position coordinates into a Gaussian potential is used based on the following formulation:

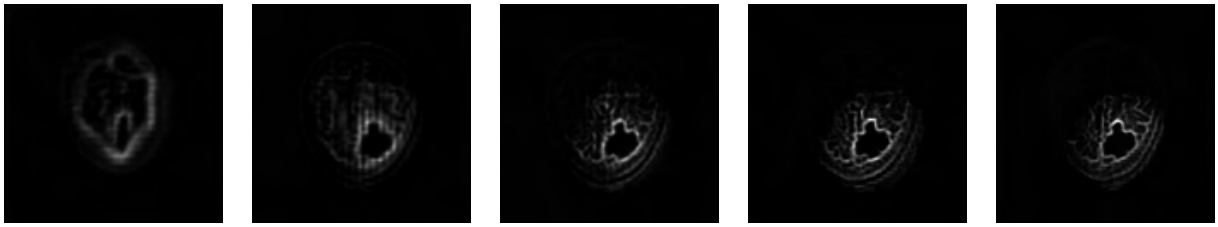
$$f(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x - b_1)^2 + (y - b_2)^2}{2\sigma^2}) \tag{5.11}$$

Here, (b_1, b_2) represents the barycenter coordinates. The predicted potential is used at inference time to identify the maximum location from which the barycenter coordinates can be extracted.

The model's architecture was initialised with 64 feature maps, a suitable number for high-resolution images, and a batch size of 16 was used during the training process. This combination of hyperparameters allowed the model to effectively use detailed information from the input image while maintaining a balance between generalisation and overfitting, as demonstrated by the results presented in this chapter. Since the two decoders are different and predict two different things, these new parameters do not assist the segmentation compared to the direct method.

5.3.3 Potential Analysis

The potential generated by the neural network was analysed with respect to the number of training epochs. Results show in Figure 5.4 that the output distribution quickly converged to-

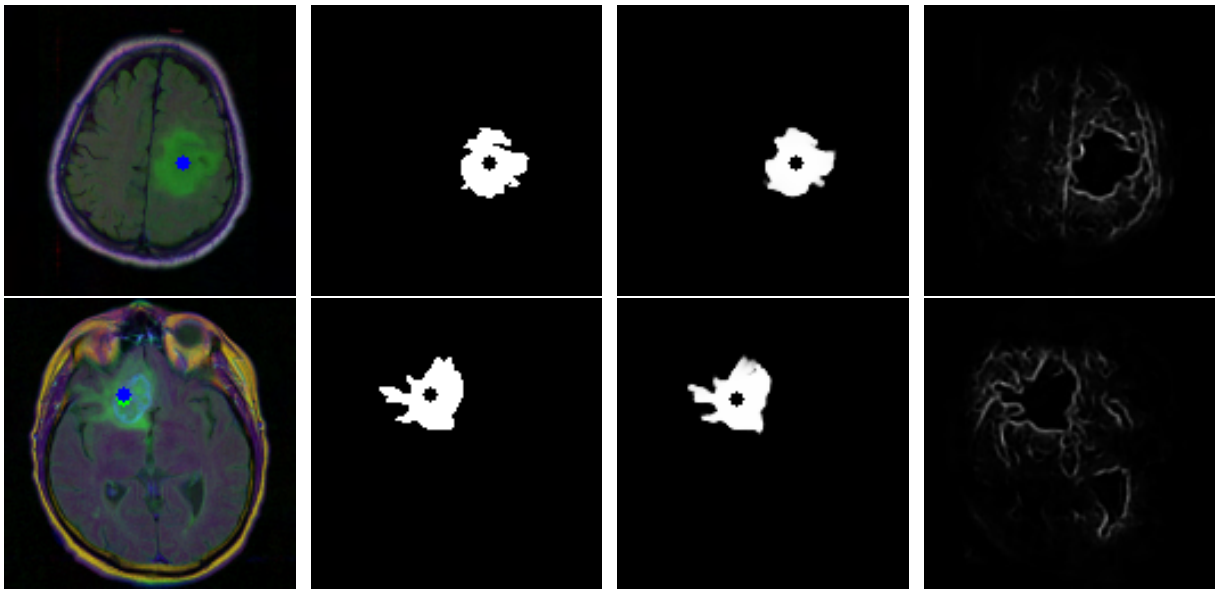


(a) Epoch 1 (b) Epoch 10 (c) Epoch 50 (d) Epoch 100 (e) Epoch 150

Figure 5.4: Evolution of the predicted potential taken as input in the Fast Marching Module.

wards the tumour's boundaries to be segmented. However, as training progressed, the contour of the tumour sharpened, and boundaries became more distinct. At the same time, we could see the brain edges removed. Ultimately, the potential only holds detailed information about the contours in a small area around the tumour.

5.3.4 Segmentation Experiments



(a) Input Image (b) Groundtruth (c) Predicted (d) Potential

Figure 5.5: Results of the segmentation on validation data. The blue and green dots on the input image are, respectively, the ground truth and predicted seed.

We compared our method to a standard U-Net segmentation approach. Our method demonstrates precise edge detection as seen in the results plots 5.5. The well-defined contours produced by our method result from its ability to take into account the morphology of the image, which traditional filters cannot do. Furthermore, our method's problem-specific nature allows for improved image segmentation performance. Classical metrics will enable us to compare quantitatively the results of our segmentation. We recover the same precision on the segmentation mask with minimal improvements in the symmetric Hausdorff distance. However, the

Table 5.1: Segmentation results (IOU) on the TGCA_LGG brain MRI database.

Name	Dice	IOU	Hausdorff	F1 Score	FPR	FNR
U-Net	0.862	0.869	2.313	0.869	0.007	0.05
ResNet U-Net	0.873	0.877	2.257	0.877	0.006	0.07
FM U-Net (ours)	0.825	0.823	2.505	0.823	0.011	0.064
FM Resnet U-Net (ours)	0.863	0.866	2.248	0.866	0.009	0.04

convergence towards an acceptable solution is faster when combined with the Fast Marching Module since, with only an approximate potential, the method converges to a relatively close segmentation. Time allows the neural network to learn the filter and sharpen the edge of the tumour. A general observation from the segmentation in Figure 5.5 is that the method, when failing to predict correctly a pixel, tends to create a false positive rather than a true negative. Table 5.1 shows our method has a high recall, controlling for very few false negatives. We performed the training with the library *HFM*. Overall, the U-Net architecture shows difficulties in precisely learning the potential. At the same time, from a metric point of view, the ResNet-U-Net performs comparatively as the classical segmentation technique using CNNs.

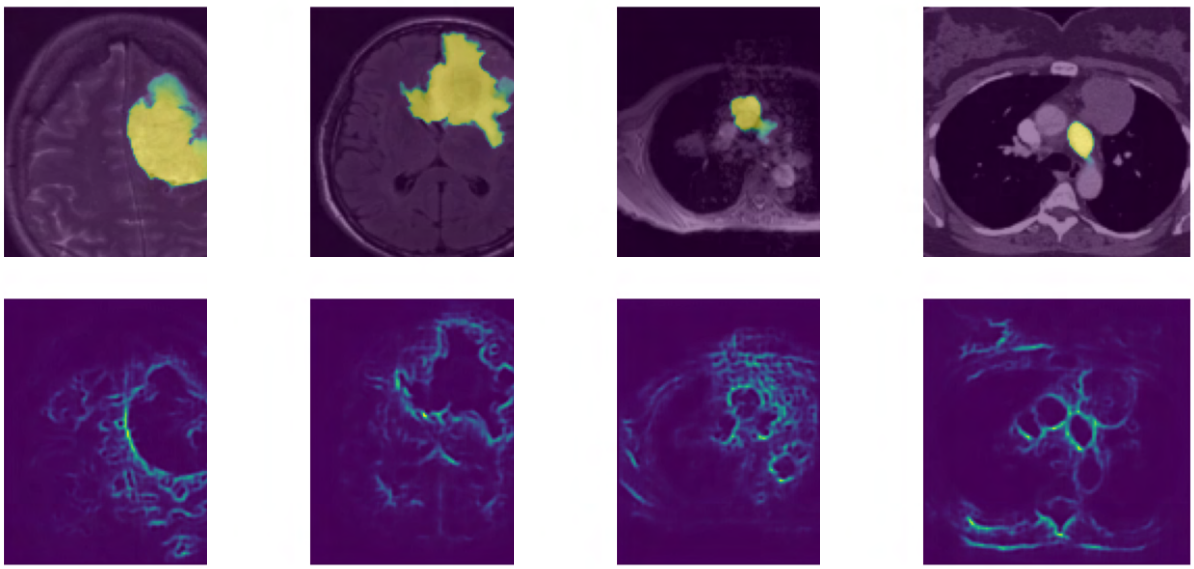


Figure 5.6: Results of the Fast Marching Energy CNN for images outside the scope of the training database.Top row: segmentation of outside the training scope. Bottom row: Potential output by the CNN before fast marching.

We also look at the properties of the generated potential of our CNN by testing it with dissimilar MRI images found randomly through an image search on Figure 5.6 where activated areas correspond to the segmentation ranging from yellow to green for confidence. The results for the last two MRI images show that while the algorithm does not properly segment the tumour (as the predicted barycenter for initialisation of the Fast Marching is not correctly placed), the learned filter detects small contours similar to tumours, focusing on the shape of

the different objects.

5.3.5 More Experimental Results

Figures 5.7[1] and 5.8[2] show examples where our trained model achieves its best and worst performances based on the F1 score.

We notice that the best scores occur when the target mask is large and looks like a simple ball. This suggests that our model performs well when the area to segment is sizable and has a straightforward shape.

In the worst cases, our model struggles because the area to be segmented is very narrow, or it fails to place a good seed point inside the area. The seed point is crucial because it's where the model starts computing the geodesic distance for segmentation.

Interestingly, it's actually easier for the model to learn how to provide a good potential for segmentation than to accurately predict the centre (barycenter) of the target segmentation mask. Predicting the barycenter is essentially projecting the mask into a two-dimensional space, which might seem simple due to its low dimensionality, but it poses challenges for the model.

We also compared two different approaches for selecting the seed point: one using the mean predicted value and the other using the maximum probability value. Using the maximum probability, the second approach reduces the number of images where our model performs very poorly. Specifically, the proportion of images with almost zero scores drops from about 7.5% to around 0.25%. Additionally, the overall performance improves from an 83% F1 score to 85%.

To provide a complete picture, we include some examples of our results on the test set in Figures 5.9[3] and 5.10[4].

[1] Figure created by Théo Bertrand and kindly provided.
[2] Figure created by Théo Bertrand and kindly provided.
[3] Figure created by Théo Bertrand and kindly provided.
[4] Figure created by Théo Bertrand and kindly provided.

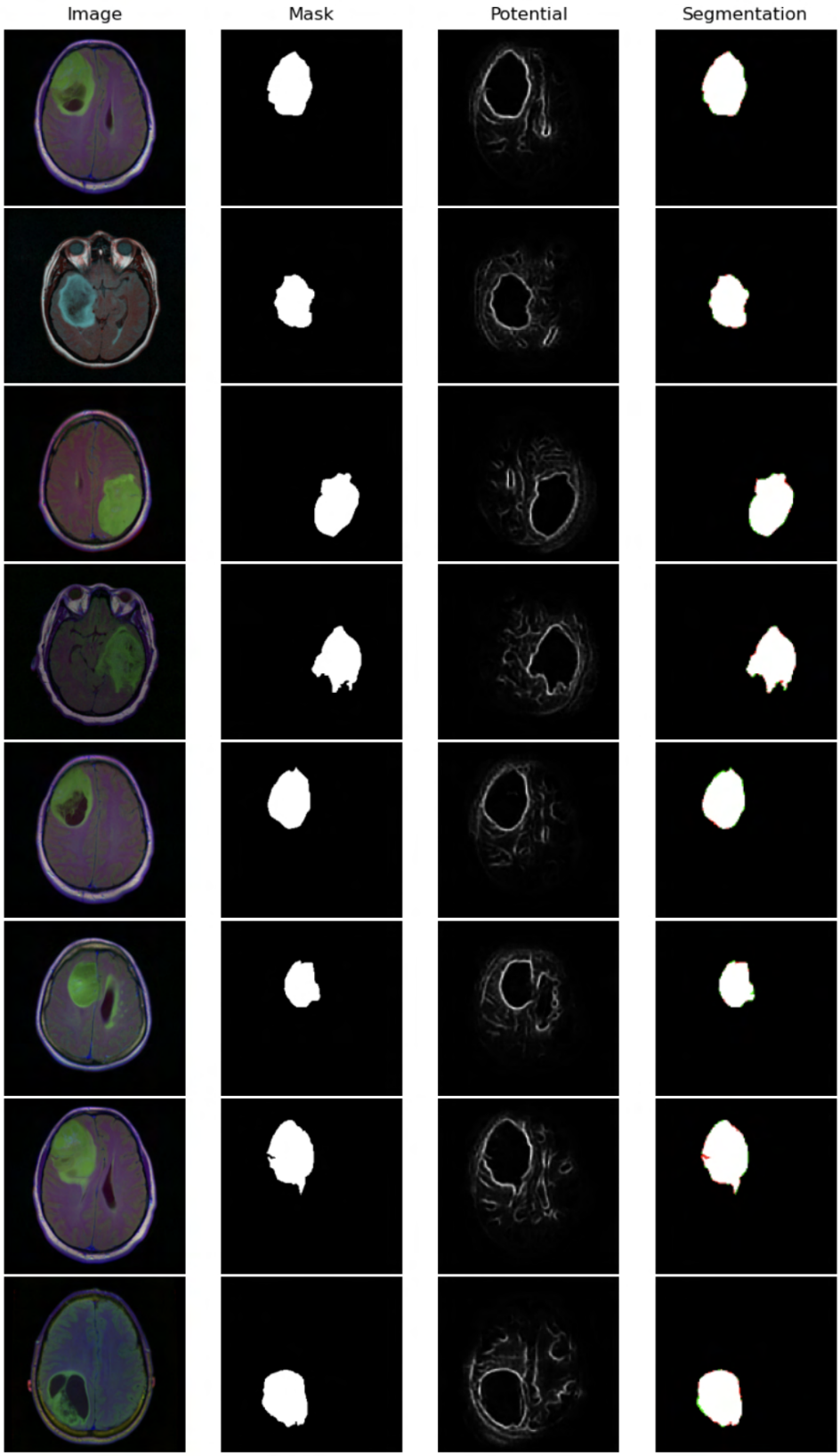


Figure 5.7: Examples where our FMECNN model achieves its highest scores. From Left to Right, the figures show the input image, the target segmentation mask, the potential computed by our trained model, and the proposed segmentation (superposed with the target, the red canal is the proposed, the green canal is the target, and blue is the intersection).

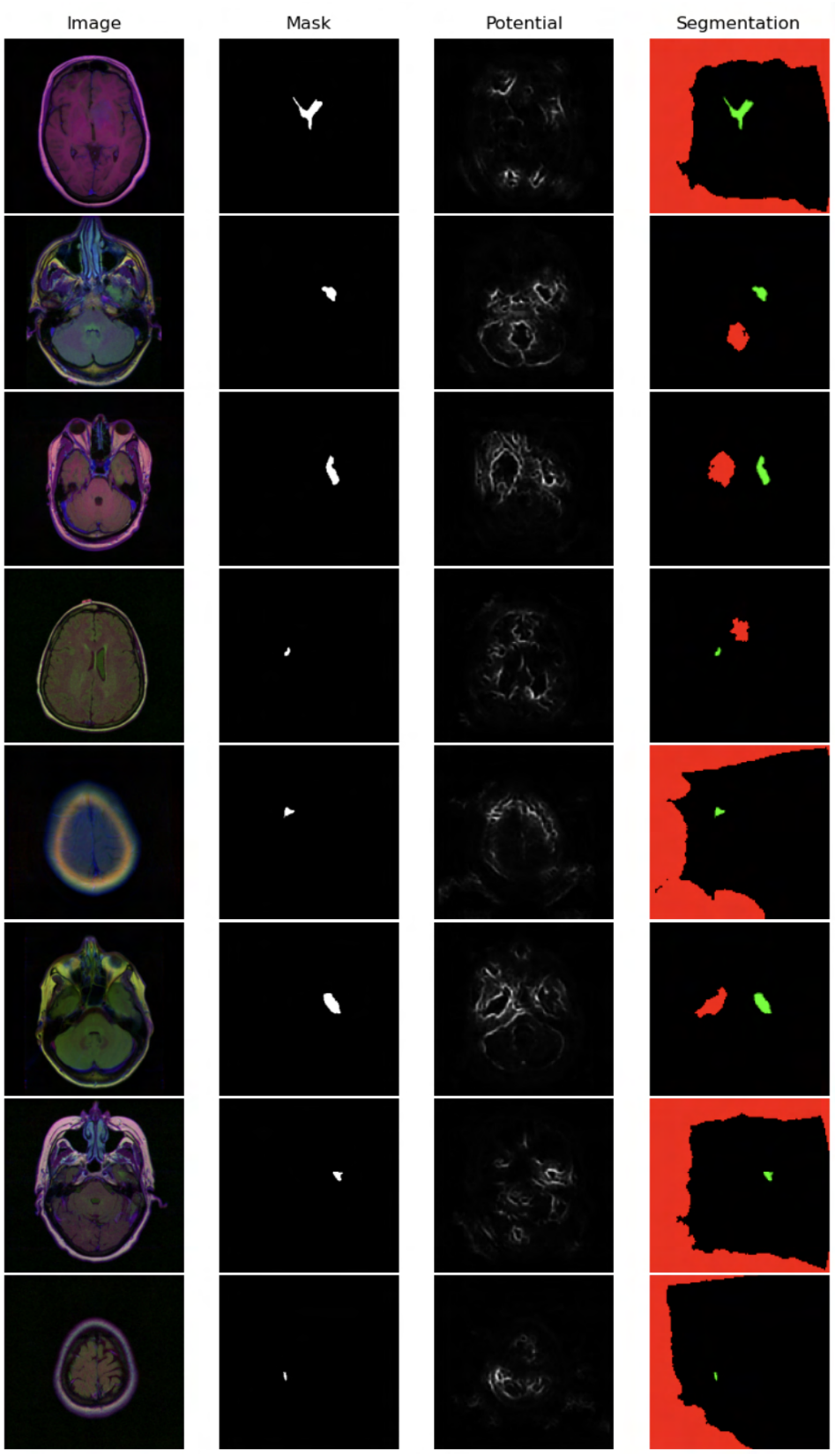


Figure 5.8: Examples where our FMECNN model achieves its lowest scores. From Left to Right, the figures show the input image, the target segmentation mask, the potential computed by our trained model, and the proposed segmentation (superposed with the target, red canal is the proposed, green canal is the target, and blue is the intersection).

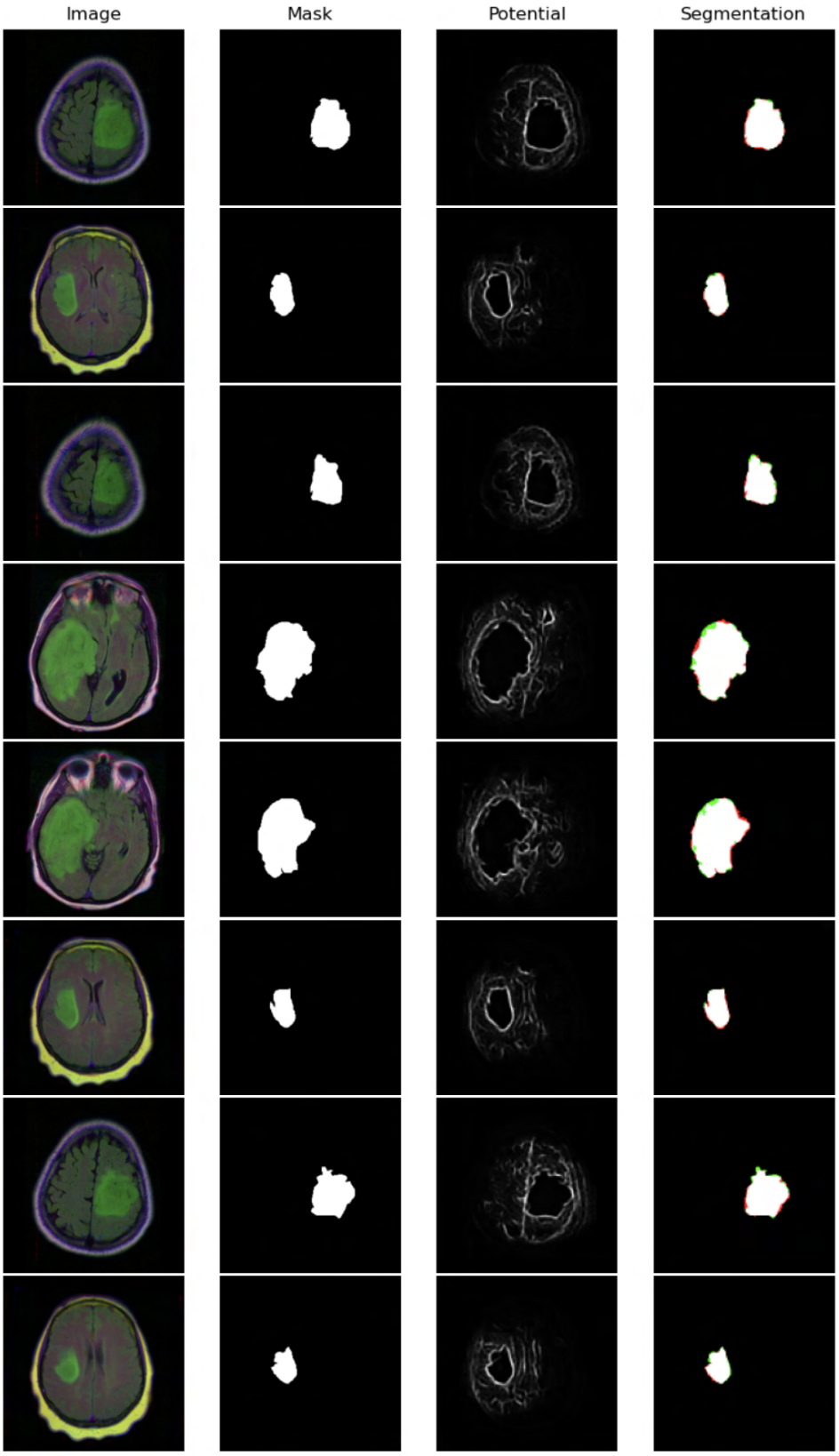


Figure 5.9: Examples where our FMECNN model achieves its highest scores on the Test set. From Left to Right, the figures show the input image, the target segmentation mask, the potential computed by our trained model, and the proposed segmentation (superposed with the target, the red canal is the proposed, the green canal is the target, and blue is the intersection).

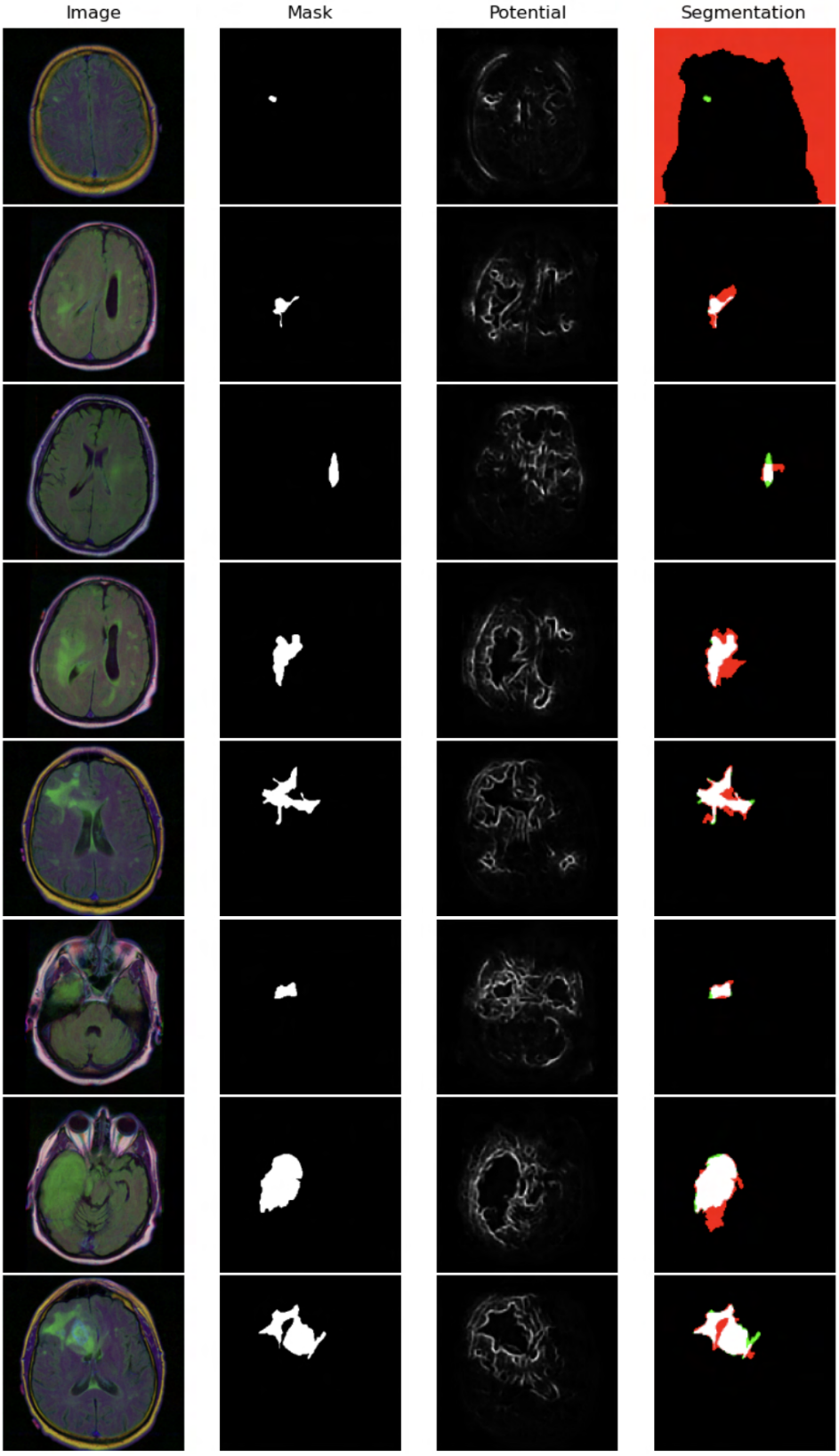


Figure 5.10: Examples where our FMECNN model achieves its lowest scores on the Test set. From Left to Right, the figures show the input image, the target segmentation mask, the potential computed by our trained model, and the proposed segmentation (superposed with the target, red canal is the proposed, green canal is the target, and blue is the intersection).

5.4 Anisotropic Geodesic Case

To compute geodesic distances, Crane et al. [Cra13] proposed an alternative approach based on the formulation introduced by Varadhan [Var67], leveraging the heat equation for efficient distance computation.

In this section, we recall the formulation of the heat equation. We begin with the simple case of an isotropic heat diffusion with a constant thermal diffusivity coefficient. Then, we recall the formulation with a thermal diffusivity coefficient depending on the grid coordinates. We move to the more technical setting of the anisotropic diffusion of heat and finally combine anisotropy and non-constant thermal diffusivity coefficient. We then present the first approach using a close method to the one presented in the isotropic section. Finally, we introduce a new model to extend the results to a more complex structure.

5.4.1 Isotropic Heat Diffusion

The heat equation is a partial differential equation that describes the propagation of heat in a medium or, more precisely, the evolution of the heat distribution over a period of time T. The general form of the equation writes:

$$\frac{\partial u}{\partial t} = \alpha \Delta u \tag{5.12}$$

where $\alpha \in \mathbb{R}_+$ is the thermal diffusivity, positive and constant in each region coordinate, and Δ represents the Laplace operator. If we position ourselves in the case where the heat propagation takes place in $\mathbb{R}^2$, then for all $x \in \mathbb{R}^2$ and for all $t \in [0, T]$, $u(x, t)$ represents the temperature at coordinates x and at time t.

The thermal diffusivity coefficient characterises the rate at which the heat diffuses through a material. Considering the thermal diffusivity α as a constant implies that the evolution of the heat distribution is the same in any medium coordinate. This is often not true in real-world scenarios since mediums rarely have homogenous properties.

The Laplace operator Δ is a second-order differential operator that plays a role in the spatial aspect of the heat equation. In $\mathbb{R}^2$, the Laplace operator takes the form $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$, representing the second-order derivatives of the heat function u with regards to the spatial coordinates. The Laplace operator captures how the temperature at a point compares to its surroundings and quantifies the local variation of the temperature. It measures the curvature of the heat distribution. If the Laplacian is positive, then the temperature at that point is smaller than its average surroundings, promoting the heat flow towards that point. We recover the physical property; the second law of thermodynamics states that heat always flows spontaneously from hotter to colder matter regions.

Changing the thermal diffusivity coefficient can make the heat equation more complex, as proposed by Yang et al. [Yan16]. We want to have a different diffusion speed at each grid coordinate. Varying the thermal diffusivity implies that the rate of heat diffusion varies spatially within the domain. Given a potential p representing the conductivity in a medium, we can write α as:

$$\alpha = |1 - |p(x_0) - p(x)||^d + \varepsilon, \forall x \in \mathbb{R}^2. \tag{5.13}$$

The point x_0 is set at the coordinates of the heat source. The coefficient $d \in \mathbb{N}^*$ will depend on the contrast between the interesting features in the image and background. The formula for $alpha$ suggests that the thermal diffusivity now differs at any domain point. The difference between the source of propagation and any other point of the domain influences it.

A higher d coefficient will result in an enhanced difference in the propagation of the heat distribution in different areas of the image. This allows for fine-tuning the sensitivity and modelling more complex propagation scenarios, such as barriers or preferred diffusion paths.

5.4.2 Anisotropic Heat Diffusion

In the previous section, we studied the propagation of heat distribution without considering the principal direction of propagation in an image. However, in many applications, it is essential to consider the direction of heat propagation. To do so, we rewrite the heat equation using a diffusion tensor, which is a generalisation of the thermal diffusivity coefficient as:

$$\frac{\partial u}{\partial t} = \text{div}(D\nabla u), \tag{5.14}$$

where D is the diffusion tensor. Usually, D is constructed using the feature of the image. The diffusion tensor allows us to control the direction of heat propagation in the image.

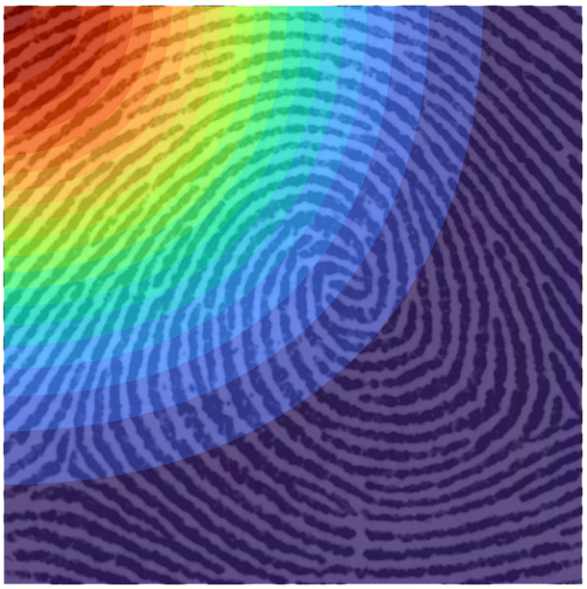
Using a diffusion tensor, we can model anisotropic diffusion, where the diffusion speed differs in different directions. Figure 5.11 illustrates the difference between isotropic and anisotropic heat propagation across a fingerprint image. Figure 5.11a depicts isotropic heat propagation, where the diffusion occurs uniformly in all directions. This type of diffusion does not consider any directional properties of the image, leading to an even distribution of heat across the surface. In contrast, Figure 5.11b, 5.11c, and 5.11d show anisotropic heat propagation, where the diffusion tensor D controls the direction of heat propagation. This allows heat to propagate faster in specific directions, depending on the image features. The contour lines show the varying anisotropic coefficients, which result in different propagation behaviours. These contours follow the principal directions of the fingerprint ridges, illustrating how the anisotropic diffusion adapts to the local geometry of the image.

Using the diffusion tensor, we can model directional heat propagation, crucial for applications requiring anisotropic diffusion to highlight specific structural features in images.

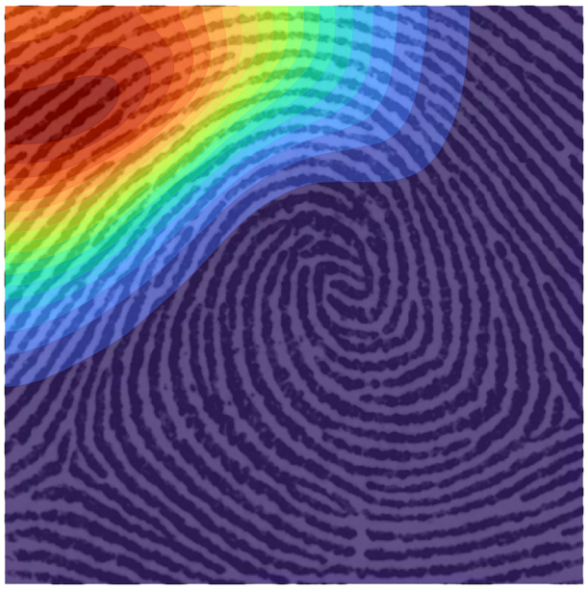
5.4.3 Structure Tensor Field

The structure tensor field T is a $2D$ matrix in each grid coordinate. It represents the orientation in a neighbourhood around a point. The idea is to measure the variation in intensity and texture between two consequently selected regions. If you have two regions with similar intensities, we want the orientation to be minor, whereas for two regions with different intensities, we want the orientation to indicate this variation.

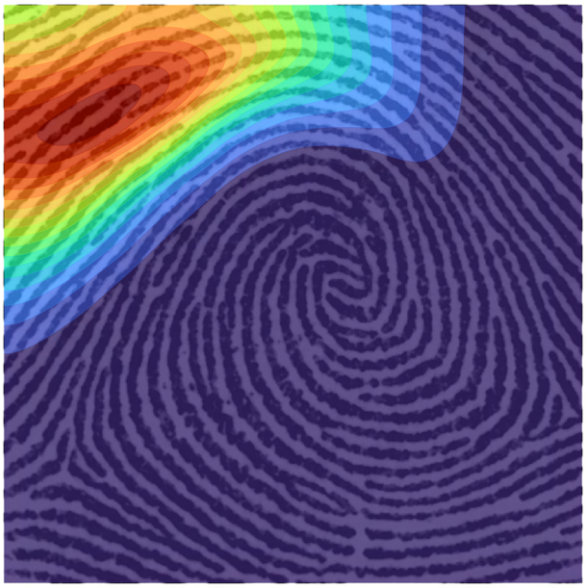
The structure tensor field is computed using the first derivatives of the image along both directions using a Gaussian filter to average information on a neighbourhood of each point to reflect the size of the neighbourhood.



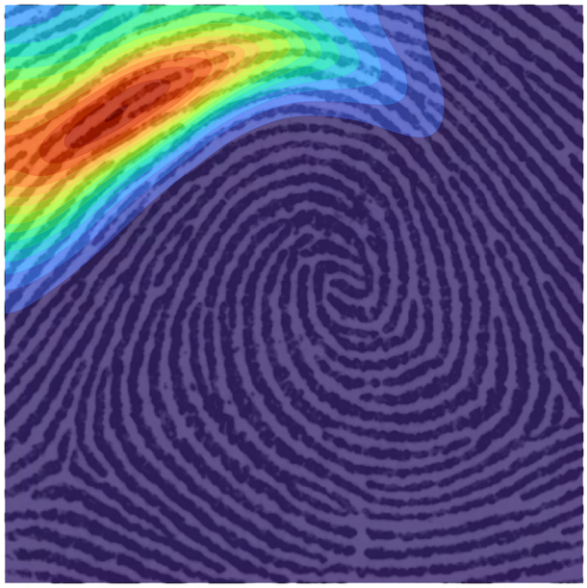
(a) Isotropic heat propagation.



(b) Anisotropic heat propagation.



(c) Anisotropic heat propagation.



(d) Anisotropic heat propagation.

Figure 5.11: The images compare isotropic (a) and anisotropic (b-d) heat propagation across a fingerprint image for different anisotropic coefficients. Isotropic diffusion spreads heat uniformly, while anisotropic diffusion, controlled by the tensor D, directs heat along the image's structural features, adapting to the fingerprint's geometry.

$$\begin{pmatrix} \frac{\partial I}{\partial x}^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial y}\frac{\partial I}{\partial x} & \frac{\partial I}{\partial y}^2 \end{pmatrix}$$

$$T = T * G_\sigma \tag{5.15}$$

The structure tensor is then diagonalised in an orthogonal basis using the eigenvectors (e_1, e_2):

$$T(x) = \mu_1(x)e_1(x)e_1(x)^T + \mu_2(x)e_2(x)e_2(x)^T \tag{5.16}$$

The eigenvectors (e_1, e_2) represent the orientation of the image at each pixel location, with e_1 indicating the direction of maximum variation and e_2 indicating the direction of minimum variation. The corresponding eigenvalues (μ_1, μ_2) represent the magnitude of the variation in each direction.

The visualisations in Figure 5.12 demonstrate how the tensor field encodes anisotropy and orientation in different datasets. The ellipses depicted in these figures correspond to the local structure tensor at each point. The direction and elongation of the ellipses provide insight into the image's dominant texture or structural direction at each location. Larger, elongated ellipses indicate areas with strong directional variation (red ellipses), where e_1 dominates. In contrast, more circular ellipses reflect regions with isotropic or minimal variation between directions (green ellipses), where μ_1 and μ_2 are closer in magnitude. This graphical representation allows an intuitive interpretation of the local geometry, revealing patterns such as flow direction or branching structures in the underlying image.

5.4.4 Varadhan Formulation

In [Var67], the author proposes a theoretical result that relates the behaviour of the heat equation to the geodesic distance on a Riemannian manifold. It was initially introduced in the context of significant deviation theory but has since found applications in various fields, such as computer vision and image processing.

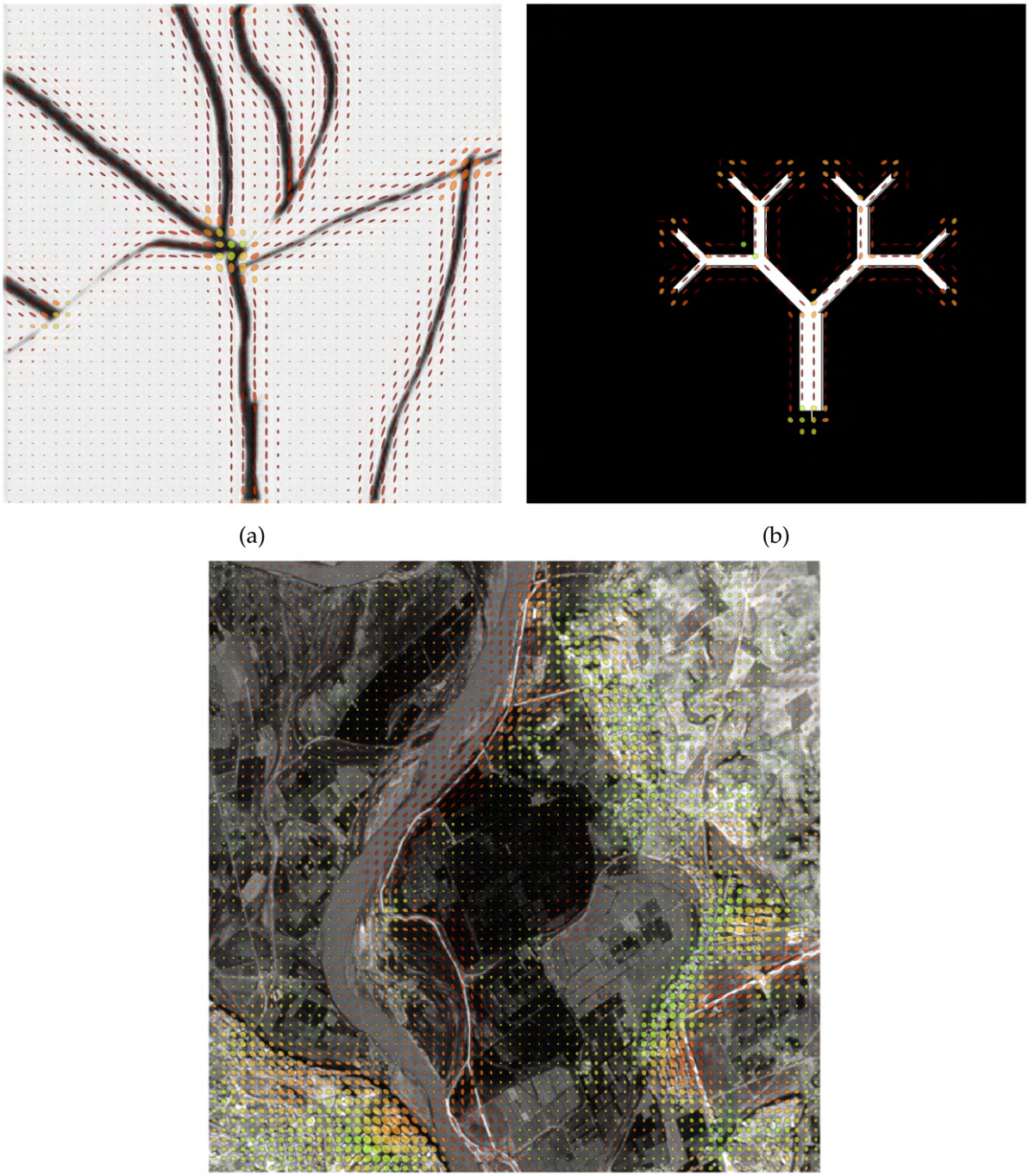
To compute the geodesic distance on an image, the favoured method uses the Fast Marching (FM) algorithm proposed by Sethian [Set96], an extension of Dijkstra's algorithm. The author Yang et al. [Yan16] adapted this result in the case of isotropic and anisotropic heat flow to get the geodesic. Their work builds on the seminal work of Crane et al. [Cra13] who proposed a method to compute the geodesic distance in three steps: i) compute the heat density $\partial_t u = \alpha \Delta u$ where α is constant on the whole domain ii) normalise the gradient $X = \nabla u / |\nabla u|$ iii) solve the Poisson equation $\Delta \phi = \text{div}(X)$ where ϕ is the final distance.

We first consider the heat equation on a Riemannian manifold:

$$\frac{\partial u}{\partial t} = \Delta u \tag{5.17}$$

where Δ is the Laplace-Beltrami operator, which generalises the Laplacian operator to curved spaces.

To apply the Varadhan formulation, they use the results introduced by on uniformly



(a) (b)

(c)

Figure 5.12: The images provide a visualisation of the anisotropy and orientation encoded in the tensor field across different datasets. The ellipses in each image represent the local anisotropy at each point, with their orientation and shape indicating the principal directions and diffusion strength. Larger and more elongated ellipses denote stronger directional anisotropy (red ellipses), while more circular ellipses suggest isotropic diffusion (green ellipses). Image (a) illustrates the synthetic dataset, with ellipses highlighting flow-like structures, emphasising the dominant diffusion directions. Finally, in (b), the tree-like structure is visualised, with ellipses following the branches, indicating the natural orientation and anisotropy within the tree structure. In (c), the road structure dataset is shown, where ellipses are aligned along the road networks, reflecting the constrained diffusion along the roads.

second-order parabolic operators with variables to approximate solutions for the Green function. The resulting Laplacian operator is defined as:

$$Lu := \sum_{i,j}^{n} a_{i,j}(x)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i,j}^{n} b_j(x)\frac{\partial u}{\partial x_j} + c(x)u. \tag{5.18}$$

Therefore, the heat equation becomes:

$$\begin{cases} \partial_t u - Lu = 0, \text{ in } (0,\infty) \times \mathbb{R}^N \\ u(x,0) = \delta_{x_0}, \text{ on } \mathbb{R}^N \end{cases}. \tag{5.19}$$

The Green function of the heat equation is a solution to the PDE with an initial condition given by a delta function at a point x_0 on the manifold. It represents the temperature distribution at time t due to an instantaneous point source. Given L according to Constantinescu et al. [Con10], the Green function for the heat equation is given by:

$$\mathcal{G}(x,y,t) = \frac{e^{-\frac{\phi(x,y)^2}{4t}}}{(4\pi t)^{N/2}}(\sum_{k=0}^{\infty} \mathcal{G}^{(k)}(x,y)) \tag{5.20}$$

According to Varadhan's formula Varadhan [Var67], the logarithm of the Green function is asymptotically equivalent to:

$$\lim_{t\to 0}[-4t \log u_x(y,t)] = \phi^2(x,y) \tag{5.21}$$

where $u_x(y,t)$ is the Green function for an initial condition given by a delta function at x and $\phi(x,y)$ is the geodesic distance between x, y induced by the Riemannian metric.

$$u_x(y,t) = (2\pi t)^{-k/2} \exp\{-\frac{1}{4t}\|x-y\|^2\} \tag{5.22}$$

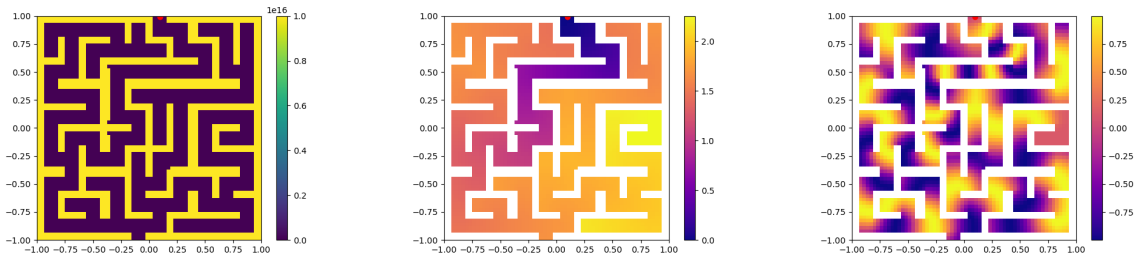
Varadhan's formula [Var67] provides a way to compute the geodesic distance between two points on a Riemannian manifold by solving the heat equation and analysing the behaviour of the Green function.

5.4.5 Numerical Applications

The methodology involves computing the Laplacian of the heat field modified to account for the anisotropic diffusion through the use of a spatially resolved diffusion tensor. This tensor modulates the diffusivity in different directions to enable a more accurate simulation of a heterogenous medium than isotropic diffusion models.

The computational procedure first computes the heat field's gradient to determine the heat changes' rate and direction within the medium. Then, a second-order derivative of the heat field is combined with the diffusion tensor to recover the anisotropic Laplacian. Additionally, the method calculates the divergence of the product between the diffusion tensor and the heat gradient.

We use Von Neumann boundary conditions. Following the CFC conditions, we carefully select the spatial and temporal discretisation parameters to ensure the method's numerical

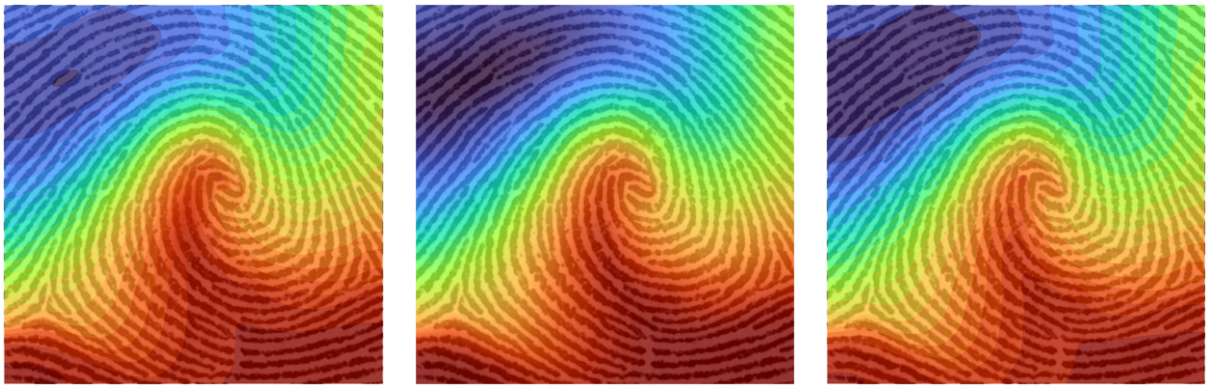


(a) (b) (c)

Figure 5.13: The images illustrate the potential field and geodesic distance computation in a maze-like environment using the Heat method. These images are to be compared with the Figure 2.12. In (a), the potential field is visualised, which serves as the input to the FM algorithm. In (b), the geodesic distance from a source point (marked in red) to all other points in the maze is shown. Lastly, in (c), the distance is modulated with a sinusoidal function to show the level-sets.

Algorithm 2 Laplacian update for the heat diffusion

function LAPLACIAN UPDATE(u, D, dt, n)
  for $i \leftarrow 1$ to n **do**
    $u \leftarrow \sum_{i,j}^{n} D_{i,j}(x) \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i,j}^{n} D_j(x) \frac{\partial u}{\partial x_j}$
    $u \leftarrow u + dt * u$
  return u



(a) (b) (c)

Figure 5.14: The plots illustrate the propagation of heat on a fingerprint image based on the anisotropic heat equation originating from a source point. In some cases, a coefficient α is defined as $\alpha = |1-|p(x_0)-p(x)||^d+\varepsilon$, which modulates the geodesic distance based on the spatial variation in heat propagation. The colour gradient, ranging from blue (short distances) to red (long distances), reflects increasing geodesic distance from the source point. In (a), the contour plot represents the anisotropic geodesic distance with the α coefficient applied, highlighting its effect on distance modulation. Plot (b) displays the raw anisotropic geodesic distance, while (c) shows the contour of the geodesic distance without including the α coefficient, comparing the two approaches.

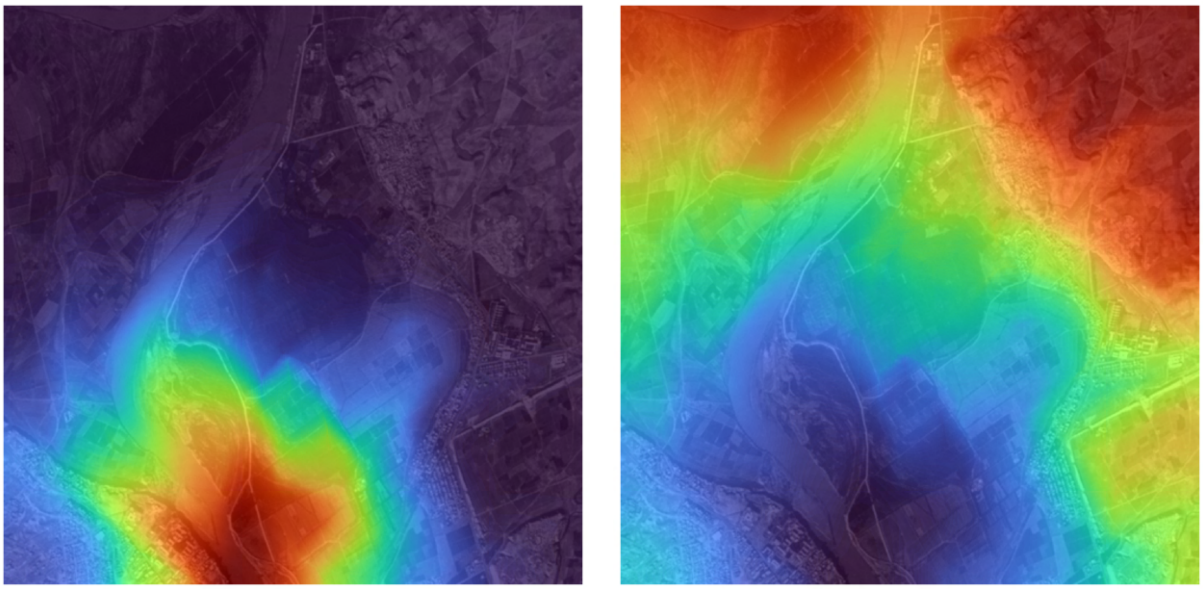
stability and convergence.

We show in Figure 5.13 the resulting distances computation on a 2D square domain with as potential a labyrinth in the case of the isotropic heat method for geodesic distance computation. In comparison, Figure 5.14 presents the results of the heat propagation also on the 2D domain but in the case of a potential defined by a fingerprint with a varying coefficient α. The effect of the coefficient α is little compared to the action of the diffusion tensor, giving rise to the anisotropic formulation of the heat equation. Figure 5.15 and 5.16 presents more results on heat propagation and anisotropic geodesic distance.

5.4.6 Generating masks with geodesic balls

In our investigation, we concentrate on enhancing the segmentation of tubular structures in images by integrating deep-learning models with geodesic distance measurements. This approach extends the principle of leveraging topological priors for reconstructing specific image regions, mainly focusing on identifying tubular structures that are inherently path-connected and exhibit a non-complex topology. Such structures are essential in various applications, including medical imaging, where blood vessels and nerves are interesting.

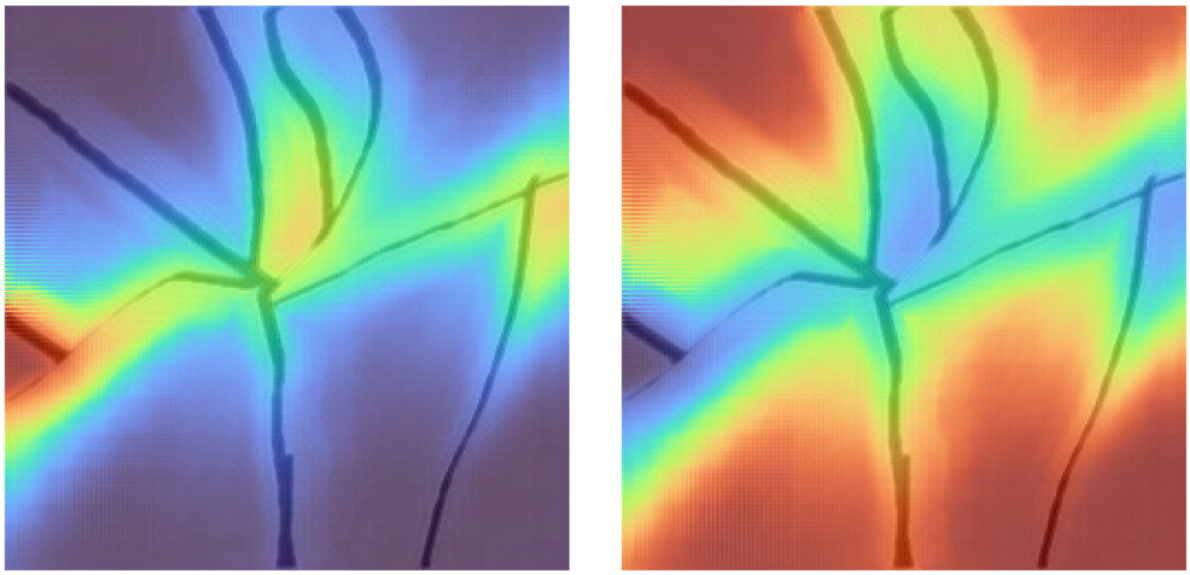
We extend the model introduced in Section 5.2.5 to anisotropic geodesic distance in order to segment tubular structures. The model predicts a potential for the Riemannian metric, subsequently applied in constructing a geodesic distance map, $d_\phi(x_0, \cdot)$. This map is crucial for delineating the tubular regions, traditionally segmented by identifying them as within a certain geodesic distance from a chosen point, x_0, in the domain $\mathbb{R}^d$. The conventional method



(a) Anisotropic heat propagation.

(b) Anisotropic geodesic distance.

Figure 5.15: The images illustrate the results of anisotropic heat propagation and geodesic distance computation. In (a), the heat propagates from a source point across the terrain, as modelled by the anisotropic heat equation. The colour gradient, ranging from blue (low temperature) to red (high temperature), represents the heat intensity at different locations, reflecting the varying resistance to heat flow. In (b), the anisotropic geodesic distance is computed, with the colour map indicating distances: blue represents regions close to the source point, while red indicates areas farther away, following the minimal paths that conform to the terrain's geometry.



(a) Isotropic heat propagation.

(b) Isotropic geodesic distance with alpha

Figure 5.16: The images illustrate the results of anisotropic heat propagation and geodesic distance computation. In (a), the heat propagates from a source point across the vascular network, as modelled by the anisotropic heat equation. The colour gradient, ranging from blue (low temperature) to red (high temperature), represents the heat intensity at different locations, reflecting the varying resistance to heat flow. In (b), the anisotropic geodesic distance is computed, with the colour map indicating distances: blue represents regions close to the source point, while red indicates areas farther away, following the minimal paths that conform to the vascular's geometry.

employs an indicator function, $\chi_{d_\phi(x_0,\cdot)\leq 1}$, for regions modelled as geometric balls within this geodesic distance framework, where $\phi \in L^1(\Omega)$ acts as the potential defining the geodesic metric.

We shift from a hard indicator function to a differentiable approximation to adapt this method for practical application in neural network architectures and ensure the optimisation process is conducive to gradient-based methods. Specifically, for the segmentation of tubular structures, the mask generation is formulated through a sigmoid function that smoothly transitions across the boundary of the tubular region. This function is defined as $\chi^\delta(d_\phi(x_0,\cdot)) = 1 - \frac{1}{1+\exp(-(d_\phi(x_0,\cdot)-1)/\delta)}$, where δ is a small positive parameter related to the pixel's size, ensuring a smooth interpolation from the interior to the exterior of the tubular region. This adaptation allows the neural network to predict a geodesic distance map that effectively segments tubular structures by approximating the characteristic function of the tubular region as $\delta \to 0$.

This method demonstrates the potential to closely model the segmentation of tubular structures by adjusting the sigmoid's steepness through δ, typically chosen about the image resolution. In the context of tubular structures, the potential ϕ is optimised to emphasise the tubular boundaries, guided by the neural network's architecture and trained using automatic differentiation and the ADAM optimiser with a learning rate of 0.01. To facilitate smooth enforcement of the potential's positivity, crucial for the fast marching method used in geodesic distance computations, we consider ϕ^2 as the input potential.

5.4.7 Learning an Anisotropic Metric

For this first work, the model extends the one presented in section 5.2.4, which presents a method that merges deep learning with the segmentation of an area as a unit ball for an anisotropic geodesic distance. We use the heat method based on Varadhan's formulation to predict a potential for the Riemannian metric and a Gaussian potential that initiates heat propagation. This approach uses the output from the model to construct an anisotropic diffusion tensor and integrate it into a geodesic distance computation module. The aim is to enhance the precision of geodesic distance measurement in heterogeneous medium, leveraging the strengths of both deep learning for feature extraction and the analytical computation of distances.

The deep learning model is the same as presented in section 5.2.4. The model keeps the same structure. Additionally, the model predicts a Gaussian potential field that serves as the source for heat propagation. This prediction is crucial for applying the heat method for geodesic distance computation, as it determines the initial conditions for the diffusion process. We use a sequence of convolutional layers with decreasing filter sizes to predict the Gaussian potential, followed by batch normalisation and ReLU activation functions. The final layer is a convolutional layer with a softmax activation function, which ensures that the output values sum to one and can be interpreted as a probability distribution.

An anisotropic diffusion tensor is constructed using the predicted potential field to accommodate a medium's directional heat flow properties. This step is crucial for accurately modelling the heat diffusion process in media where thermal properties vary with direction. The anisotropic diffusion tensor allows for a more nuanced heat flow simulation, reflecting the complex behaviour of tubular structures.

Finally, we solve the heat equation approaching the time-zero limit, and Varadhan's formulation bridges the gap between the heat kernel and the geodesic distance. This process yields a geodesic distance map.

In segmenting tubular structures within images, our methodology advances the distance computation module to accommodate the unique characteristics of tubular geometries and the anisotropic properties of their surrounding media. The module operates on the premise that the geodesic distance, denoted as $d_\phi(x_0, \cdot)$, calculates the proximity from any given point on the grid to predefined seed points, x_0, based on an anisotropic metric, ϕ. This metric, ϕ, significantly diverges from traditional isotropic metrics by encapsulating directional dependencies, essential for accurately modelling the complex pathways within tubular structures.

For scenarios involving multiple seed points, indicative of tubular networks with branching paths or intersections, the model seamlessly accommodates an array of seeds, $S = \{x_0^i\}_{1 \leq i \leq q}$. Here, the distance to the closest seed, $d\phi(S, \cdot) = \min_{x_0^i \in S} d_\phi(x_0^i, \cdot)$, is computed, ensuring that the model's applicability spans across tubular networks of varied complexity. This computation does not impose additional burdens on the fast marching algorithm or the heat method employed to deduce these distances efficiently.

The definition of the metric ϕ stems from the output of a neural network specifically tailored for medical imaging and the segmentation of tubular structures. The U-Net architecture, renowned for its effectiveness in medical image processing, is the backbone for our neural network, denoted as f_θ where $\theta \in \mathbb{R}^p$ represents the parameter space. In our model, ϕ is defined as $f_\theta(u)^2 + \epsilon$, with u representing the input image. This formulation ensures that the metric ϕ remains positive and non-zero, a critical attribute for the accurate computation of anisotropic geodesic distances.

5.4.8 Experiments

5.4.8.1 Data

For our investigation, we leveraged three distinct datasets encompassing both synthetic and real-world medical imaging data to evaluate the robustness and applicability of our models. Figure 5.17 presents an example of each dataset.

We use a synthetic dataset comprising images of small trees designed to test the model's capability in detecting arboreal structures. This set includes 20 images for training and 20 for testing. Points within these images, representing critical junctures in the tree structure, are stored as coordinates. For each image, we generated Gaussian heatmaps centred around these key points to facilitate the learning of structural nuances and initialise the heat propagation.

Another synthetic dataset was curated to simulate simple vascular structures, containing 40 images for training and 20 for testing. Like the small trees dataset, key points are identified as critical vessel junctures and represented through Gaussian heatmaps. This dataset aims to benchmark the model's performance in recognising and segmenting linear and branching patterns typical of vascular networks.

We used the Multi-Atlas Labeling Beyond the Cranial Vault dataset, which contains full-resolution images. This dataset provides detailed ground truth labels for semantic segmentation across various anatomical structures. The dataset's high-resolution nature allows for

(a) (b) (c)

Figure 5.17: The images represent examples from different datasets used for the different segmentation tasks. (a) The first image depicts a medical CT scan, highlighting anatomical structures relevant to segmentation in medical imaging. (b) The second image shows a synthetic tree structure used for evaluating segmentation algorithms on geometrically complex patterns. (c) The third image illustrates a detailed vascular tree structure, focusing on the segmentation of fine, branching elements.

identifying intricate features within the images, such as organ boundaries and other important anatomical structures.

Ground truth labels are used to train the segmentation models across all datasets. We aim to develop models capable of accurately segmenting and outlining key anatomical features in synthetic datasets and real-world medical images. The dataset is carefully split into independent training and test sets to ensure a robust and unbiased evaluation of the model's segmentation performance.

We used data augmentation techniques to increase the diversity of the synthetic datasets and improve the model's ability to generalise. These techniques include horizontal and vertical flipping, random rotations, and geometric transformations like shifting and scaling. Using these strategies, the model is exposed to a wider variety of structural orientations and scales, which is important for achieving high accuracy in keypoint detection and heatmap generation.

The DRIVE patches dataset was prepared with annotations of bifurcation points, endpoints, and crossing points, adding real-world complexity not present in the synthetic datasets. We ensure comprehensive learning and adaptability by training the segmentation models on simple to complex structures.

5.4.8.2 Training Procedures

This study uses the U-Net architecture for image segmentation. The model was initialised and optimised exactly as the isotropic model presented in section 5.3.2.

5.4.8.3 Results

The images presented in this section demonstrate the segmentation results achieved by our model under different conditions. In Figure 5.18, we apply heat propagation to segment tubu-

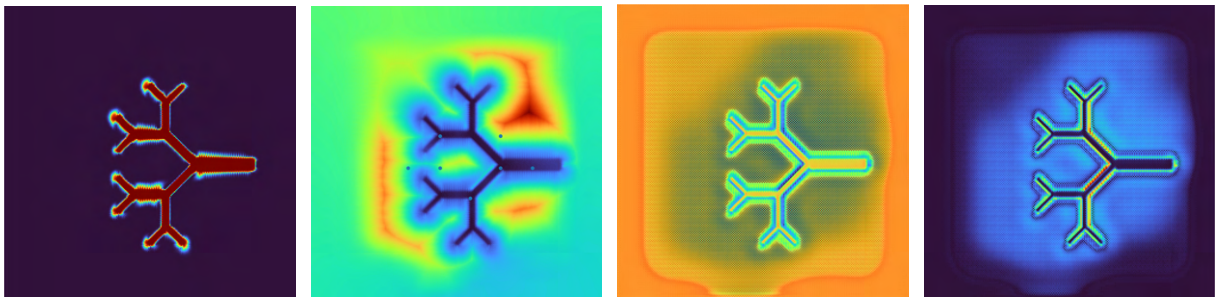


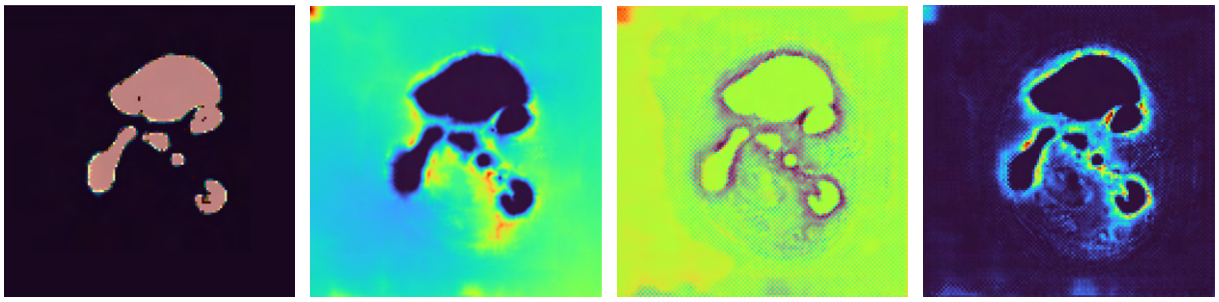
Figure 5.18: The images demonstrate the segmentation results for a tree-like structure using a heat propagation model, attempting to replicate the performance of the Fast Marching Energy CNN (FMECNN) model. In (a), the predicted segmentation mask is shown, capturing the general structure of the tree. Image (b) illustrates the predicted learned isotropic metric used to guide the segmentation process. In (c), the output of the model is visualised, highlighting the effects of anisotropic diffusion on the tree structure. Lastly, (d) displays the potential field predicted by the model, showing the influence of the learned metric in shaping the geodesic distance and segmentation.

lar structures. The heat-based method effectively outlines the vascular structures by propagating from a source point within the object. While the segmentation result is accurate, slight edge inaccuracies are noticeable, particularly where the boundaries exhibit fine, tubular geometries.

In Figure 5.19, we extend the method to handle multiple objects with similar characteristics. Here, the segmentation targets are more complex, resembling the objects typically used in the Fast Marching algorithm. The segmentation map is reasonably precise, although edge details are slightly blurred, especially in areas where object components are close together. The neural network successfully predicts the anisotropic metric, as shown by the learned potential maps, which effectively capture the object boundaries.

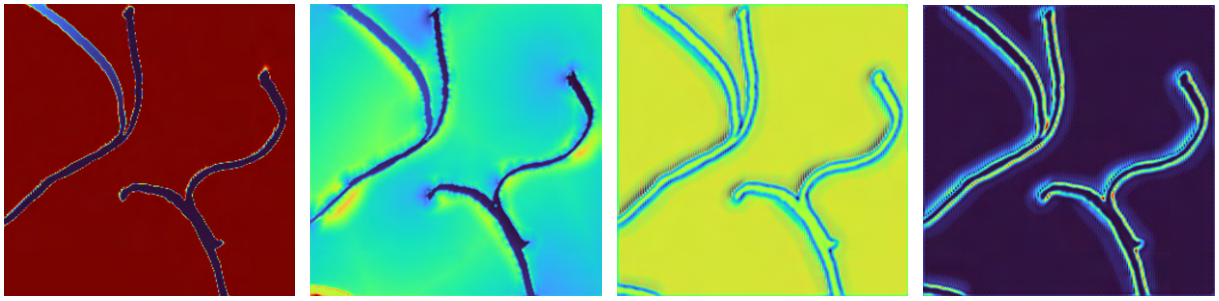
Next, in Figure 5.20 and Figure 5.21, we introduce anisotropy through the structure tensor in the learning phase. These figures illustrate the impact of anisotropic diffusion on the segmentation process. The introduction of anisotropy allows the model to better handle directional features in the data, particularly in structures with complex geometries such as branching trees and tubular vessels. The predicted segmentation masks in these figures align well with the intricate shapes of the objects, especially in cases where previous isotropic methods struggled.

The potential maps in these figures further demonstrate the model's ability to predict geodesic distances from a source point inside the object to its boundary. The anisotropic nature of the model allows for more accurate boundary delineation, which is critical for applications involving complex geometries like vascular or branching structures. Despite some minor imperfections at the edges of the segmented regions, the overall results are promising, indicating that anisotropy helps improve segmentation quality in datasets involving tubular or tree-like structures and in synthetic and real-world scenarios.



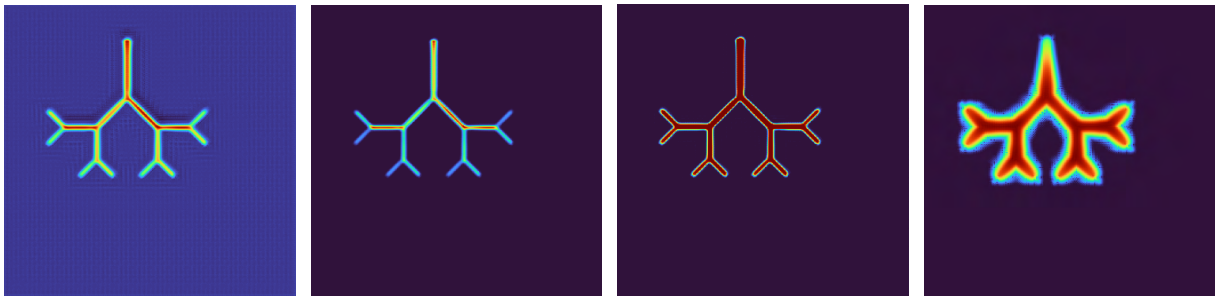
(a) (b) (c) (d)

Figure 5.19: The images show the results of a semantic segmentation task using a neural network to predict an anisotropic metric. In the first image (a), the model's output prediction is presented, showcasing the segmented region. Image (b) displays the predicted anisotropic metric the neural network learned. This metric influences the diffusion process. In (c), the model's output on the segmented structure is visualised, showing the effects of anisotropic diffusion. Finally, image (d) illustrates the predicted potential field, showing the progression of diffusion within the segmented region. These results demonstrate the model's capacity to learn and apply the predicted metric for accurate semantic segmentation.



(a) (b) (c) (d)

Figure 5.20: The images display the results of synthetic data analysis using anisotropic diffusion. In the first image (a), the model's output prediction is shown, demonstrating the segmentation result of the synthetic structure. Image (b) presents the predicted anisotropic metric guiding the behaviour of the diffusion process. In (c), the final output of the diffusion model on the synthetic structure is visualised. Lastly, image (d) illustrates the predicted potential field, showing how the diffusion progresses through the synthetic structure. These results highlight the model's ability to capture and represent complex synthetic features.



(a) (b) (c) (d)

Figure 5.21: The images demonstrate the results of anisotropic tree structure analysis. In the first image (a), the output of the anisotropic model on the tree structure is shown, illustrating the heat propagation pattern. In (b), the potential field generated from the anisotropic diffusion is visualised, highlighting how the heat flows through the tree branches. Image (c) shows the modulation of the geodesic distance with the coefficient α, the thermal diffusion, affecting the diffusion process based on local variations. Finally, in (d), the predicted segmentation or outline of the tree structure is displayed, highlighting the model's ability to capture the geometric features of the tree.

5.4.9 Learning an Anisotropic Metric - Another Approach

In contrast to the initial approach described above, we present a second method for geodesic distance computation within vascular networks, addressing several challenges encountered with seed point selection and computational efficiency. Specifically, this approach avoids the complexity of manually selecting a "seed" point from which to compute the distance map—an issue made difficult by the non-convex nature of the vascular regions to be segmented and the absence of a unique optimal point. For this second approach, we approach the Laplacian operator as the graph Laplacian on the grid as Heitz et al. [Hei21].

Initially, we considered leveraging a database of points of interest, similar to our previous work on landmark detection and geodesic fitting. Using these predefined seed points, we achieved the segmentation results depicted in Figure 5.22. The geodesic distance was computed using the heat equation based on Varadhan's formulation. However, while computing geodesic distances from a single seed point is relatively straightforward, challenges arise when distances must be calculated from multiple points. Our approach aimed to recover regions to segment, denoted $y_{\text{th}} \in \{0,1\}^n$, by approximating:

$$\exp\left(-\frac{d_M(x_0,x_i)^2}{4t}\right) \approx (y_{\text{th}})_i \tag{5.23}$$

where x_0 is a seed point, and x_i represents a grid point. A significant complication emerged: although the segmented region around each seed point is normalised to sum to 1, the mass distribution around each seed may differ, leading to inconsistencies in the segmentation. To overcome this, we would compute a non-trivial renormalisation constant for each region or solve the heat equation individually for each seed, taking the minimum across all distance maps. Both options are computationally expensive.

To simplify this, we propose a different strategy that avoids explicitly predicting seed points

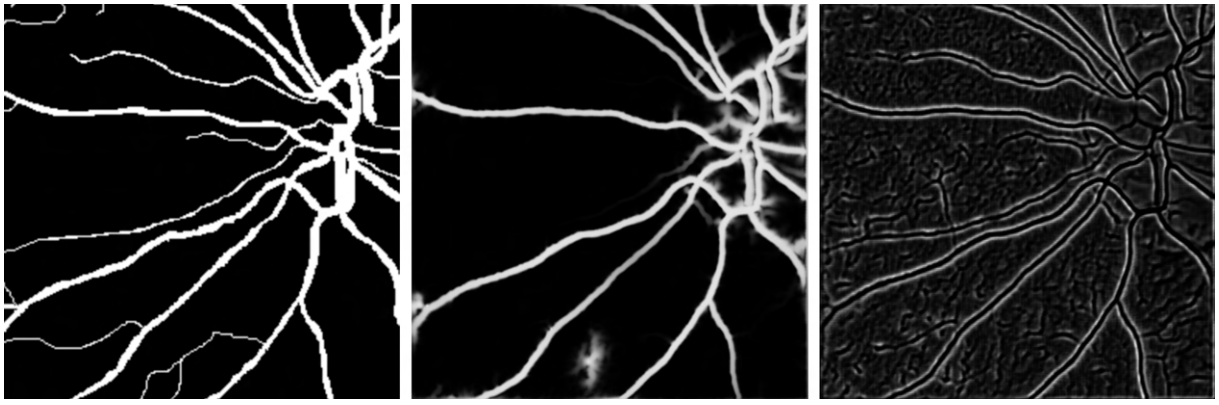


Figure 5.22: First example of generating an isotropic metric with the help of a U-Net on an image from the validation set. Left: Ground Truth Segmentation. Center: Proposed Segmentation. Right: Associated Potential output by the U-Net

from the image. Instead, we modify the initialisation of the heat flow by replacing the position of δ_x, the Dirac delta function representing the seed location. This allows us to predict the heat flow without specifying exact seed points.

In this alternative approach, our neural network is composed of two branches (See Figure 5.23). The first branch predicts the map $x \mapsto D(x)$, where $D(x)$ defines the heat flow and is the inverse of the classic metric tensor. The second branch predicts a $2D$ probability map μ, which indicates likely vascular landmarks or regions of interest that guide the heat flow and refine the segmentation.

Given input images x and their corresponding normalised ground truth masks y, we define our loss function as:

$$\mathcal{L}_{\text{seg}}(\Phi_t^{D_\theta(x)}(\mu_\theta(x)), y) - \lambda \|\mu_\theta(x)\|_2^2 \tag{5.24}$$

where D_θ is the metric tensor predicted by the first branch, μ_θ is the probability map from the second branch, and $\Phi_t^D(\nu)$ represents the heat flow based on diffusion tensor D, applied to $\nu \in P(\Omega)$ until time t .

We trained this network on the DRIVE dataset [Sta04], splitting it into a 60% training set and 40% validation set, and tested it on the IOSTAR dataset [Zha16]. To prevent overfitting, we applied random affine transformations and image flips, using only the green channel of the input images. We found that a small batch size of 2 images yielded better results. After 250 epochs, the model achieved a DICE score of 77% on the validation set, which, although not state-of-the-art, effectively overcomes some of the challenges identified earlier (see Figure 5.22).

As illustrated in Figure 5.24, the barycenter predicted by our model does not coincide with the barycenter maps produced by the original FMECNN model. Instead, it predicts segmentations of the vascular network. This divergence occurs because the second branch of our network is not explicitly trained to predict seed points, and the sparsity-promoting penalty in the loss function does not collapse the barycenter to a set of discrete points, even for larger values of λ. This behaviour can be interpreted as an attention mechanism, where the solution to the

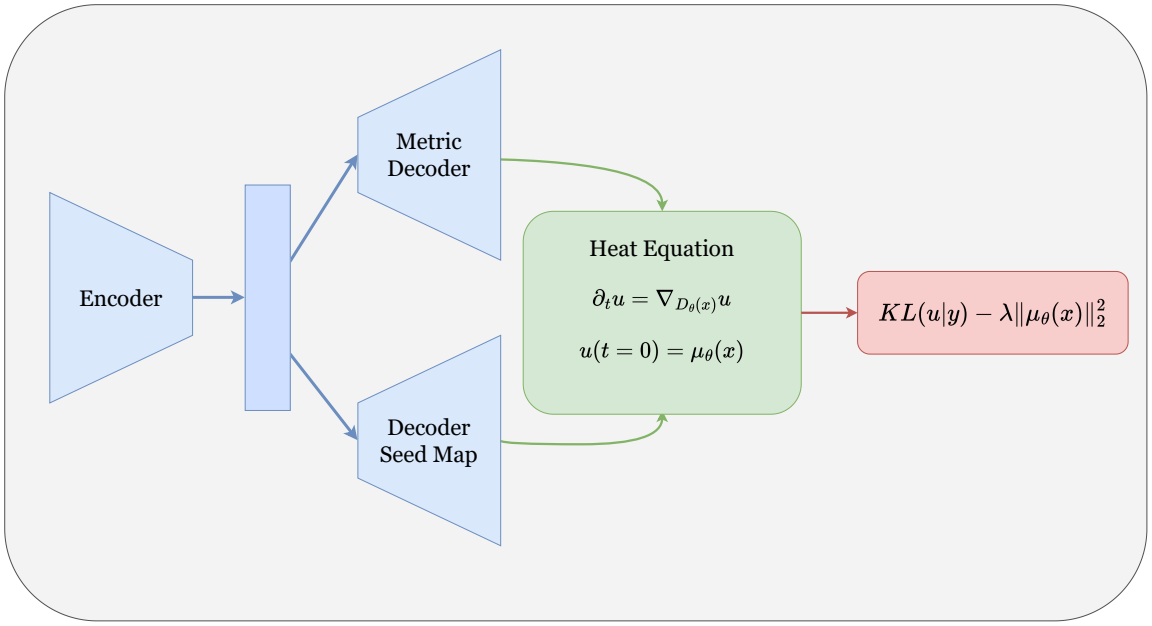


Figure 5.23: Diagram of the architecture of the alternative approach for anisotropic tubular structure segmentation.

heat equation, computed from the two network outputs, refines the segmentation map.

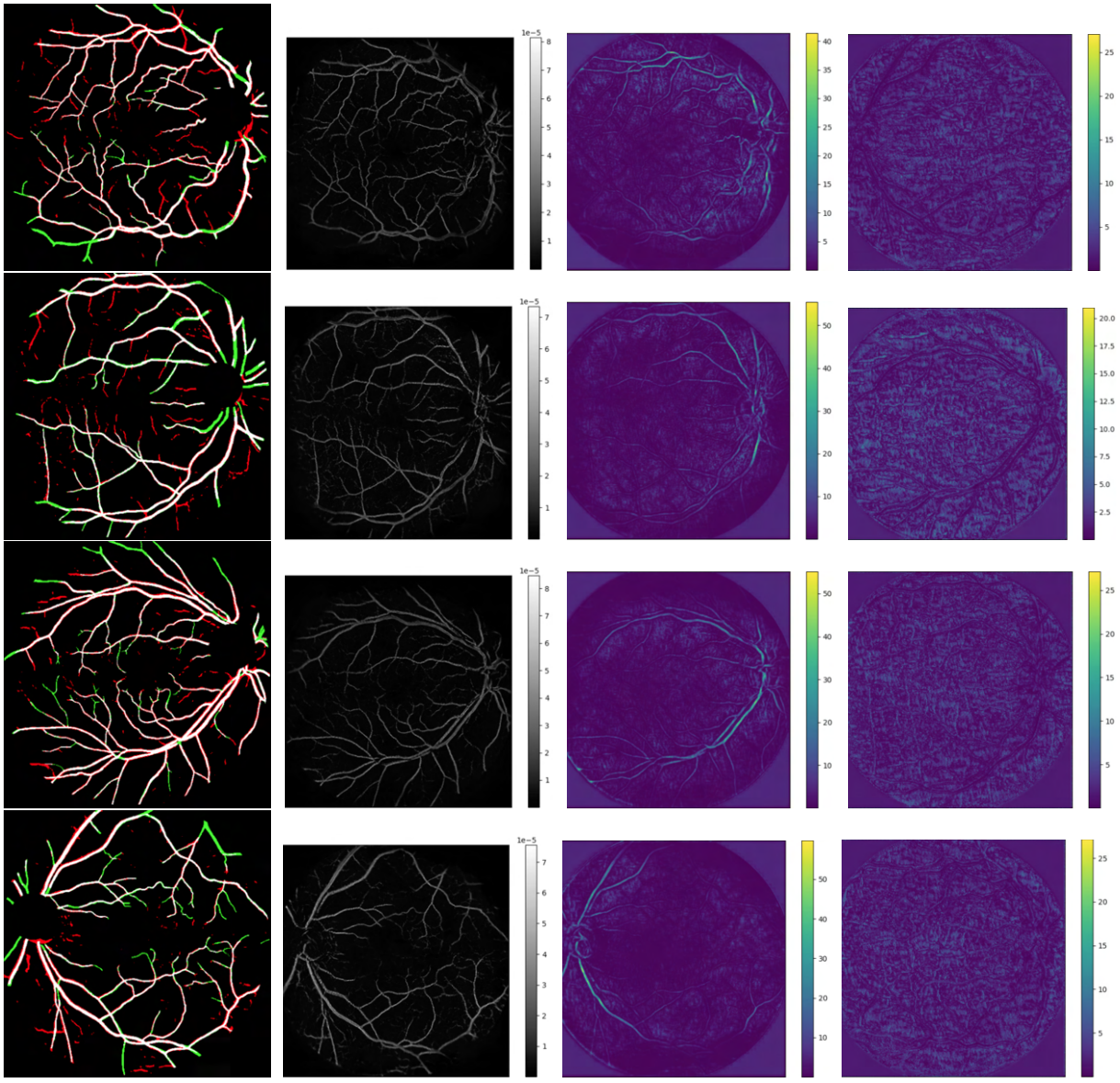


Figure 5.24: Output of our method on a sample from the IOSTAR dataset. Left: Comparison of proposed segmentation versus Ground Truth. Center Left: Barycen- ter map output by the network. Center Right: Sum of the metric elements in both directions. Right : (log of) Anisotropy factor.

5.5 Partial Conclusion

In this chapter, we introduced a novel approach to image segmentation that combines the Fast Marching Method (FMM) with Convolutional Neural Networks (CNNs), termed the Fast Marching Energy CNN (FMECNN). Our primary goal was to leverage the rich geometrical information provided by geodesic distances to enhance segmentation tasks, particularly in medical imaging applications such as brain tumour segmentation.

We began by exploring the isotropic geodesic case, where the metric tensor is uniform in all directions. Integrating the Fast Marching algorithm into a neural network framework allowed the network to learn the potential function ϕ directly from the data. This potential function serves as the speed map for the Fast Marching algorithm, enabling the computation of geodesic distances from a source point. We addressed the challenge of differentiating through the Fast Marching algorithm by using the Subgradient Marching Algorithm, allowing us to backpropagate gradients through the geodesic distance computations.

Our method involved predicting both the potential function and the seed point from which to compute the geodesic distances. The segmentation mask was then generated by thresholding the geodesic distance map, effectively defining the segmented region as a geodesic ball centred at the predicted seed point. We employed a smooth approximation of the characteristic function of the geodesic ball using a sigmoid function, which facilitated gradient-based optimisation during training.

Experimental results on the TCGA LGG brain MRI database demonstrated that our FMECNN model could achieve segmentation performance comparable to traditional U-Net architectures. Notably, our method produced precise edge detections and maintained robustness against false negatives, a critical factor in medical image segmentation where missing a region of interest can have significant consequences. The potential function learned by the network effectively captured the boundaries of the tumours, highlighting the model's ability to incorporate geometric information into the segmentation process.

We also extended our approach to the anisotropic geodesic case to handle more complex image structures, such as tubular networks in vascular imaging. By incorporating anisotropic diffusion tensors derived from the structure tensor field of the image, we adapted the heat equation-based geodesic distance computation to account for directional information. This extension allowed the model to capture the intrinsic geometries of anisotropic structures better, leading to improved segmentation results in datasets involving tree-like and vascular structures.

Two models were proposed for the anisotropic case. The first model extended the isotropic approach by integrating anisotropic diffusion into the heat equation and predicting the diffusion tensor and the seed points. The second model circumvented the need for explicit seed point prediction by employing a probability map and Kullback-Leibler (KL) regularisation, effectively transforming the segmentation problem into an attention mechanism guided by the heat flow.

Our experiments with these models showcased their ability to segment complex structures without the necessity of predefined seed points, reducing computational overhead and simplifying the segmentation pipeline. Although the models did not surpass state-of-the-art perfor-

mance metrics, they provided valuable insights into integrating geometric methods with deep learning for image segmentation.

In conclusion, the FMECNN framework presents a promising direction for incorporating geometric priors into deep learning-based segmentation models. By leveraging geodesic distances and the Fast Marching Method within a neural network architecture, we can impose geometric and topological constraints on the output masks, leading to more accurate and reliable segmentation results. This approach opens up new possibilities for applications where the shape and connectivity of the segmented regions are of paramount importance, such as in medical imaging and computational anatomy.

Future work could focus on further refining the anisotropic models to enhance their performance and extend their applicability to three-dimensional datasets. Additionally, exploring the integration of other geometric PDE-based methods within deep learning frameworks could yield new insights and methodologies for complex image analysis tasks. Addressing computational efficiency and scalability will also be essential for deploying these models in real-world clinical settings.

Conclusions

6.1 Conclusion

This work explores new ways to improve image segmentation by combining traditional mathematical methods with modern Deep Learning techniques, especially for medical images. Our main goal was to make segmentation models more accurate and reliable by directly including geometric and topological information in neural networks.

First, we introduced the **Chan-Vese Attention Gate**, an attention mechanism added to the U-Net model. This method includes the Chan-Vese energy minimisation inside the network's attention gates, allowing better control over the segmentation masks. This means the network can focus more effectively on important regions, like tumours in medical images, which leads to better segmentation accuracy. By integrating the Chan-Vese method, the network can learn from the data while keeping the geometric constraints from the energy minimisation.

Our experiments showed that the Chan-Vese Attention Gate helps the network to outline complex structures in images more effectively. The attention masks created by our model focused on critical regions, confirming that our approach is good at capturing the spatial information needed for segmentation. This method improved the Intersection over Union (IoU) scores compared to standard U-Net models and reduced false negatives, which is very important in medical diagnostics where missing a region of interest can have serious consequences.

Next, we introduced the **Fast Marching Energy CNN (FMECNN)**, which combines the Fast Marching Method with Convolutional Neural Networks to use geodesic distances in image segmentation. The Fast Marching Method is a numerical algorithm used to solve the Eikonal equation, which describes how a wavefront moves through a medium. By integrating this method into a neural network, we allowed the network to learn the potential function directly from the data, which acts as the speed map for the Fast Marching algorithm.

We looked at both isotropic and anisotropic cases. In the isotropic case, where the metric is the same in all directions, we showed how the network could segment regions defined as geodesic balls based on the learned metric. By using a smooth approximation of the characteristic function of the geodesic ball, we made gradient-based optimisation during training possible. Our experiments showed that the FMECNN could achieve segmentation performance similar to traditional U-Net architectures, with precise edge detections and robustness against false negatives.

In the anisotropic case, we extended our approach to handle more complex image struc-

tures, like tubular networks in vascular imaging. By incorporating anisotropic diffusion tensors derived from the image's structure tensor field, we adapted the heat equation-based geodesic distance computation to account for directional information. This allowed the model to capture the shapes of anisotropic structures better, leading to improved segmentation results in datasets involving tree-like and vascular structures.

Our methods show a strong connection between heat flow and geodesic distances. By simulating how heat would naturally spread through an image's features, we could find meaningful distances that help understand and segment complex structures. This approach builds on Crane et al.'s method by integrating it into a learnable framework suitable for deep learning.

Our work shows that combining traditional mathematical methods with deep learning can enhance image segmentation tasks. We can achieve more accurate and reliable segmentation results by embedding geometric and topological constraints into neural network architectures. This is especially important in medical imaging applications, where accurately identifying structures like tumours or vascular networks is crucial for diagnosis and treatment planning.

In the future, we could focus on further improving these models, extending them to handle higher-dimensional data, and exploring more ways to integrate geometric information into deep learning frameworks. By continuing to bridge the gap between classical methods and modern machine learning, we aim to develop tools that can better understand and analyse complex image data, ultimately helping advancements in medical diagnostics and other fields that rely on accurate image segmentation.

Liste des publications

Articles à comité de lecture

Nicolas Makaroff, Laurent D. Cohen. "*Chan-Vese Attention U-Net: An Attention Mechanism for Robust Segmentation*". International Conference on Geometric Science of Information.
DOI: doi.org/10.1007/978-3-031-38299-4_59

Nicolas Makaroff, Théo Bertrand, Laurent D. Cohen. "*Fast Marching Energy CNN*". International Conference on Scale Space and Variational Methods in Computer Vision.
DOI: doi.org/10.1007/978-3-031-31975-4_21

Articles en préparation

Nicolas Makaroff, Théo Bertrand, Laurent D. Cohen. "*Learning Anisotropic Metrics for Geodesic Distances via the Heat Equation for Image Segmentation*". International Conference on Scale Space and Variational Methods in Computer Vision.

Conférences internationales

Nicolas Makaroff, Laurent D. Cohen. "*Image Segmentation Using Chan-Vese Energy Minimization Coupled with a CNN Provided Mask*". SIAM Conference on Imaging Science (IS24), Atlanta, Georgia, USA, 2024.

Nicolas Makaroff, Théo Bertrand, Laurent D. Cohen. "*Region Segmentation Defined As the Unit Ball for the Geodesic Distance with Respect to a CNN Generated Riemannian Metric* ". SIAM Conference on Imaging Science (IS24), Atlanta, Georgia, USA, 2024.

List of references

[Akb22] Parastoo Akbarimoghaddam, Atefeh Ziaei, and Hamed Azarnoush. "Deep Active Contours Using Locally Controlled Distance Vector Flow". *Signal, Image and Video Processing* 16.7 (2022), pp. 1773–1781 (cit. on p. 80).

[All02] Grégoire Allaire, François Jouve, and Anca-Maria Toader. "A Level-Set Method for Shape Optimization". *Comptes Rendus. Mathématique* 334.12 (2002), pp. 1125–1130 (cit. on p. 32).

[Alz21] Yahya Alzahrani and Boubakeur Boufama. "Biomedical Image Segmentation: A Survey". *SN Computer Science* 2.4 (2021), p. 310 (cit. on p. 12).

[Amb21] L. Ambrosio, E. Brué, and D. Semola. *Lectures on Optimal Transport*. UNITEXT. Springer International Publishing, 2021 (cit. on p. 112).

[Bad17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495 (cit. on p. 74).

[Bad08] Noor Badshah and Ke Chen. "Multigrid Method for the Chan-Vese Model in Variational Segmentation". *Communications in Computational Physics* 4.2 (2008), pp. 294–316 (cit. on p. 38).

[Bae09] Egil Bae and Xue-Cheng Tai. "Efficient Global Minimization for the Multiphase Chan-Vese Model of Image Segmentation". *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer. 2009, pp. 28–41 (cit. on p. 38).

[Bar96] Eric Bardinet, Laurent Cohen, and Nicholas Ayache. "Tracking Medical 3D Data With a Deformable Parametric Model". *Computer Vision—ECCV'96: 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings, Volume I 4*. Springer. 1996, pp. 315–328 (cit. on p. 20).

[Ben10] Fethallah Benmansour, Guillaume Carlier, Gabriel Peyré, and Filippo Santambrogio. "Derivatives With Respect to Metrics and Applications: Subgradient Marching Algorithm". *Numerische Mathematik* 116 (2010), pp. 357–381 (cit. on pp. 6, 18, 112, 114–116).

[Ber10] Alex Berg, Jia Deng, and L Fei-Fei. *Large Scale Visual Recognition Challenge 2010*. 2010 (cit. on p. 71).

[Ber23] Théo Bertrand, Nicolas Makaroff, and Laurent D Cohen. "Fast Marching Energy CNN". *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2023, pp. 276–287 (cit. on pp. 5, 17).

[Bon20] Matthieu Bonnivard, Elie Bretin, and Antoine Lemenant. "Numerical Approximation of the Steiner Problem in Dimension 2 and 3". *Mathematics of Computation* 89.321 (2020), pp. 1–43 (cit. on p. 114).

[Bor06] Folkmar Bornemann and Christian Rasch. "Finite-Element Discretization of Static Hamilton-Jacobi Equations Based on a Local Variational Principle". *Computing and Visualization in Science* 9 (2006), pp. 57–69 (cit. on p. 41).

[Can86] John Canny. "A Computational Approach to Edge Detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (1986), pp. 679–698 (cit. on p. 29).

[Car20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-End Object Detection With Transformers". *European conference on computer vision*. Springer. 2020, pp. 213–229 (cit. on p. 73).

[Cas97] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. "Geodesic Active Contours". *International Journal of Computer Vision* 22 (1997), pp. 61–79 (cit. on pp. 27, 28, 48, 83).

[Cha01] Tony F Chan and Luminita A Vese. "Active Contours Without Edges". *IEEE Transactions on Image Processing* 10.2 (2001), pp. 266–277 (cit. on pp. 3, 15, 22, 35, 38, 49, 83).

[Che14] Da Chen, Laurent D Cohen, and Jean-Marie Mirebeau. "Vessel Extraction Using Anisotropic Minimal Paths and Path Score". *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2014, pp. 1570–1574 (cit. on pp. 5, 17).

[Che16] Da Chen, Jean-Marie Mirebeau, and Laurent D Cohen. "Vessel Tree Extraction Using Radius-Lifted Keypoints Searching Scheme and Anisotropic Fast Marching Method". *Journal of Algorithms & Computational Technology* 10.4 (2016), pp. 224–234 (cit. on pp. 5, 17, 111).

[Che18] Da Chen, Jiong Zhang, and Laurent D Cohen. "Minimal Paths for Tubular Structure Segmentation With Coherence Penalty and Adaptive Anisotropy". *IEEE Transactions on Image Processing* 28.3 (2018), pp. 1271–1284 (cit. on pp. 5, 17).

[Che19] Xu Chen, Bryan M Williams, Srinivasa R Vallabhaneni, Gabriela Czanner, Rachel Williams, and Yalin Zheng. "Learning Active Contour Models for Medical Image Segmentation". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 11632–11640 (cit. on p. 82).

[Çiç16] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. "3D U-Net: Learning Dense Volumetric Segmentation From Sparse Annotation". *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer. 2016, pp. 424–432 (cit. on p. 74).

[Coh90] L Cohen. "A Finite Element Method Applied to New Active Contour Models and 3D Reconstruction From Cross Sections". *Proc. 3rd Int. Conf. on Computer Vision*. 1990, pp. 587–591 (cit. on p. 20).

[Coh91] Laurent D Cohen. "On Active Contour Models and Balloons". *CVGIP: Image Understanding* 53.2 (1991), pp. 211–218 (cit. on pp. 26, 27, 48).

[Coh96] Laurent D Cohen. "Deformable Surfaces and Parametric Models to Fit and Track 3D Data". *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929)*. Vol. 4. IEEE. 1996, pp. 2451–2456 (cit. on p. 20).

[Coh93a] Laurent D Cohen, Eric Bardinet, and Nicholas Ayache. "Surface Reconstruction Using Active Contour Models". PhD thesis. INRIA, 1993 (cit. on pp. 20, 22).

[Coh93b] Laurent D Cohen and Isaac Cohen. "Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.11 (1993), pp. 1131–1147 (cit. on pp. 29, 31).

[Coh01] Laurent D Cohen and Thomas Deschamps. "Multiple Contour Finding and Perceptual Grouping as a Set of Energy Minimizing Paths". *Energy Minimization Methods in Computer Vision and Pattern Recognition: Third International Workshop, EMMCVPR 2001 Sophia Antipolis, France, September 3–5, 2001 Proceedings 3*. Springer. 2001, pp. 560–575 (cit. on p. 20).

[Coh07] Laurent D Cohen and Thomas Deschamps. "Segmentation of 3D Tubular Objects With Adaptive Front Propagation and Minimal Tree Extraction for 3D Medical Imaging". *Computer Methods in Biomechanics and Biomedical Engineering* 10.4 (2007), pp. 289–305 (cit. on p. 111).

[Coh97] Laurent D Cohen and Ron Kimmel. "Global Minimum for Active Contour Models: A Minimal Path Approach". *International Journal of Computer Vision* 24 (1997), pp. 57–78 (cit. on pp. 111, 114).

[Con10] Radu Constantinescu, Nick Costanzino, Anna L Mazzucato, and Victor Nistor. "Approximate Solutions to Second Order Parabolic Equations. I: Analytic Estimates". *Journal of Mathematical Physics* 51.10 (2010) (cit. on p. 132).

[Cra13] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. "Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow". *ACM Transactions on Graphics (TOG)* 32.5 (2013), pp. 1–11 (cit. on pp. 127, 130).

[Dal05] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893 (cit. on p. 71).

[Del95] Philippe Delagnes, Jenny Benois, and Dominique Barba. "Active Contours Approach to Object Tracking in Image Sequences With Complex Background". *Pattern Recognition Letters* 16.2 (1995), pp. 171–178 (cit. on p. 20).

[DIJ59] EW DIJKSTRA. "A Note on Two Problems in Connexion With Graphs". *Numerische Mathematik* 1 (1959), pp. 269–271 (cit. on p. 41).

[Dos20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, et al. "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale". *arXiv Preprint arXiv:2010.11929* (2020) (cit. on pp. 70, 73).

[EZ07] Noha El Zehiry, Steve Xu, Prasanna Sahoo, and Adel Elmaghraby. "Graph Cut Optimization for the Mumford-Shah Model". *The Seventh IASTED International Conference on Visualization, Imaging and Image Processing*. 2007, pp. 182–187 (cit. on p. 38).

[Eps87] CL Epstein and Michael Gage. "The Curve Shortening Flow". *Wave Motion: Theory, Modelling, and Computation: Proceedings of a Conference in Honor of the 60th Birthday of Peter D. Lax*. Springer. 1987, pp. 15–59 (cit. on p. 34).

[Fis36] R. A. Fisher. *Iris*. UCI Machine Learning Repository. 1936 (cit. on p. 56).

[Fuj93] Kouta Fujimura, Naokazu Yokoya, and Kazuhiko Yamamoto. "Motion Tracking of Deformable Objects by Active Contour Models Using Multiscale Dynamic Programming". *Journal of Visual Communication and Image Representation* 4.4 (1993), pp. 382–391 (cit. on p. 20).

[Fuk80] Kunihiko Fukushima. "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position". *Biological Cybernetics* 36.4 (1980), pp. 193–202 (cit. on p. 70).

[Ger19] Sarah E Gerard and Joseph M Reinhardt. "Pulmonary Lobe Segmentation Using A Sequence of Convolutional Neural Networks For Marginal Learning". *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE. 2019, pp. 1207–1211 (cit. on p. 79).

[Get12] Pascal Getreuer. "Chan-Vese Segmentation". *Image Processing on Line* 2 (2012), pp. 214–224 (cit. on p. 38).

[Gir15] R Girshick. "Fast R-Cnn". *arXiv Preprint arXiv:1504.08083* (2015) (cit. on p. 74).

[Gir14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587 (cit. on pp. 53, 74).

[Gir12] Ross B Girshick, Pedro F Felzenszwalb, and David McAllester. "Discriminatively Trained Deformable Part Models, Release 5" (2012) (cit. on p. 53).

[Goo14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, et al. "Generative Adversarial Nets". *Advances in Neural Information Processing Systems* 27 (2014) (cit. on p. 74).

[Gra20] Mara Graziani, Vincent Andrearczyk, Stéphane Marchand-Maillet, and Henning Müller. "Concept Attribution: Explaining CNN Decisions to Physicians". *Computers in Biology and Medicine* 123 (2020), p. 103865 (cit. on p. 11).

[Hat19] Ali Hatamizadeh, Assaf Hoogi, Debleena Sengupta, Wuyue Lu, Brian Wilcox, Daniel Rubin, et al. "Deep Active Lesion Segmentation". *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10*. Springer. 2019, pp. 98–105 (cit. on p. 80).

[Hav17] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, et al. "Brain Tumor Segmentation With Deep Neural Networks". *Medical Image Analysis* 35 (2017), pp. 18–31 (cit. on p. 74).

[He15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving Deep Into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification". *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034 (cit. on p. 118).

[He16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 71, 116).

[He07] Lin He and Stanley Osher. "Solving the Chan-Vese Model by a Multiphase Level Set Algorithm Based on the Topological Derivative". *Scale Space and Variational Methods in Computer Vision: First International Conference, SSVM 2007, Ischia, Italy, May 30-June 2, 2007. Proceedings 1*. Springer. 2007, pp. 777–788 (cit. on p. 38).

[Hei21] Matthieu Heitz, Nicolas Bonneel, David Coeurjolly, Marco Cuturi, and Gabriel Peyré. "Ground Metric Learning on Graphs". *Journal of Mathematical Imaging and Vision* 63 (2021), pp. 89–107 (cit. on pp. 5, 17, 112, 142).

[Iso17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. "Image-to-Image Translation With Conditional Adversarial Networks". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134 (cit. on p. 74).

[Jaf16] Mohammad H Jafari, Nader Karimi, Ebrahim Nasr-Esfahani, Shadrokh Samavi, S Mohamad R Soroushmehr, K Ward, et al. "Skin Lesion Segmentation in Clinical Images Using Deep Learning". *2016 23rd International conference on pattern recognition (ICPR)*. IEEE. 2016, pp. 337–342 (cit. on p. 74).

[Kar19] Christina Karam, Kenjiro Sugimoto, and Keigo Hirakawa. "Fast Convolutional Distance Transform". *IEEE Signal Processing Letters* 26.6 (2019), pp. 853–857 (cit. on p. 46).

[Kas88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active Contour Models". *International Journal of Computer Vision* 1.4 (1988), pp. 321–331 (cit. on pp. 20, 22, 48).

[Kim21] Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. "XProtoNet: Diagnosis in Chest Radiography With Global and Local Explanations". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 15719–15728 (cit. on p. 11).

[Kin14] Diederik P Kingma. "Adam: A Method for Stochastic Optimization". *arXiv Preprint arXiv:1412.6980* (2014) (cit. on p. 118).

[Kle16] Jens Kleesiek, Gregor Urban, Alexander Hubert, Daniel Schwarz, Klaus Maier-Hein, Martin Bendszus, et al. "Deep MRI Brain Extraction: A 3D Convolutional Neural Network for Skull Stripping". *NeuroImage* 129 (2016), pp. 460–469 (cit. on p. 79).

[Koh20] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, et al. "Concept Bottleneck Models". *International conference on machine learning*. PMLR. 2020, pp. 5338–5348 (cit. on p. 11).

[Kri09] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning Multiple Layers of Features From Tiny Images" (2009) (cit. on p. 74).

[Kri12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet Classification With Deep Convolutional Neural Networks". *Advances in Neural Information Processing Systems* 25 (2012) (cit. on pp. 53, 71).

[LaL20] Rodney LaLonde, Drew Torigian, and Ulas Bagci. "Encoding Visual Attributes in Capsules for Explainable Medical Diagnoses". *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*. Springer. 2020, pp. 294–304 (cit. on p. 11).

[Le21] Ngan Le, Toan Bui, Viet-Khoa Vo-Ho, Kashu Yamazaki, and Khoa Luu. "Narrow Band Active Contour Attention Model for Medical Segmentation". *Diagnostics* 11.8 (2021), p. 1393 (cit. on p. 82).

[LeC89] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, et al. "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Computation* 1.4 (1989), pp. 541–551 (cit. on pp. 63, 70).

[Led17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690 (cit. on p. 74).

[Ley93] Frederic Leymarie and Martin D. Levine. "Tracking Deformable Objects in the Plane Using an Active Contour Model". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.6 (1993), pp. 617–634 (cit. on p. 20).

[Lim13] Joseph J Lim, C Lawrence Zitnick, and Piotr Dollár. "Sketch Tokens: A Learned Mid-Level Representation for Contour and Object Detection". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 3158–3165 (cit. on p. 53).

[Lon15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440 (cit. on pp. 74, 79).

[Los17] Ilya Loshchilov, Frank Hutter, et al. "Fixing Weight Decay Regularization in Adam". *arXiv Preprint arXiv:1711.05101* 5 (2017) (cit. on p. 99).

[Low99] David G Lowe. "Object Recognition From Local Scale-Invariant Features". *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157 (cit. on p. 71).

[Ma20] Jun Ma, Jian He, and Xiaoping Yang. "Learning Geodesic Active Contours for Embedding Object Global Information in Segmentation CNNs". *IEEE Transactions on Medical Imaging* 40.1 (2020), pp. 93–104 (cit. on p. 83).

[Mak23] Nicolas Makaroff and Laurent D Cohen. "Chan-Vese Attention U-Net: An Attention Mechanism for Robust Segmentation". *International Conference on Geometric Science of Information*. Springer. 2023, pp. 574–582 (cit. on pp. 3, 15).

[Mal98] Ravi Malladi and James A Sethian. "A Real-Time Algorithm for Medical Shape Recovery". *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE. 1998, pp. 304–310 (cit. on pp. 5, 17, 111).

[Mal95] Ravi Malladi, James A Sethian, and Baba C Vemuri. "Shape Modeling With Front Propagation: A Level Set Approach". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.2 (1995), pp. 158–175 (cit. on p. 32).

[Mar18] Diego Marcos, Devis Tuia, Benjamin Kellenberger, Lisa Zhang, Min Bai, Renjie Liao, et al. "Learning Deep Structured Active Contours End-to-End". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8877–8885 (cit. on pp. xv, 80, 81).

[McC43] Warren S McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity". *The Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133 (cit. on p. 63).

[McI00] Tim McInerney and Demetri Terzopoulos. "T-Snakes: Topology Adaptive Snakes". *Medical Image Analysis* 4.2 (2000), pp. 73–91 (cit. on p. 25).

[Mil16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". *2016 fourth international conference on 3D vision (3DV)*. Ieee. 2016, pp. 565–571 (cit. on p. 74).

[Mir19] Jean-Marie Mirebeau and Jorg Portegies. "Hamiltonian Fast Marching: A Numerical Solver for Anisotropic and Non-Holonomic Eikonal PDEs". *Image Processing on Line* 9 (2019), pp. 47–93 (cit. on p. 114).

[Mum89] David Bryant Mumford and Jayant Shah. "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems". *Communications on Pure and Applied Mathematics* (1989) (cit. on pp. 4, 15, 35, 36).

[Net11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. "Reading Digits in Natural Images With Unsupervised Feature Learning". *NIPS workshop on deep learning and unsupervised feature learning*. Vol. 2011. 2. Granada. 2011, p. 4 (cit. on p. 74).

[Nor14] Alireza Norouzi, Mohd Shafry Mohd Rahim, Ayman Altameem, Tanzila Saba, Abdolvahab Ehsani Rad, Amjad Rehman, et al. "Medical Image Segmentation Methods, Algorithms, and Applications". *IETE Technical Review* 31.3 (2014), pp. 199–213 (cit. on p. 12).

[Okt18] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, et al. "Attention U-Net: Learning Where to Look for the Pancreas". *arXiv Preprint arXiv:1804.03999* (2018) (cit. on pp. xvi, 91, 95, 96, 102).

[Osh88] Stanley Osher and James A Sethian. "Fronts Propagating With Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations". *Journal of Computational Physics* 79.1 (1988), pp. 12–49 (cit. on pp. 32, 48).

[Par02] Nikos Paragios and Rachid Deriche. "Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation". *International Journal of Computer Vision* 46 (2002), pp. 223–247 (cit. on p. 22).

[Par18] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, et al. "Image Transformer". *International conference on machine learning*. PMLR. 2018, pp. 4055–4064 (cit. on p. 73).

[Ped] N Pedano, AE Flanders, L Scarpace, T Mikkelsen, JM Eschbacher, B Hermes, et al. "The Cancer Genome Atlas Low Grade Glioma Collection (TCGA-LGG)(version 3)(2016)". *DOI: Https://doi. Org/10.7937 K* 9 () (cit. on pp. 4, 17, 83, 99, 118).

[Per18] Sérgio Pereira, Raphael Meier, Victor Alves, Mauricio Reyes, and Carlos A Silva. "Automatic Brain Tumor Grading From MRI Data Using Convolutional Neural Networks and Quality Assessment". *Understanding and Interpreting Machine Learning in Medical Image Computing Applications: First International Workshops, MLCN 2018, DLF 2018, and iMIMIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16-20, 2018, Proceedings 1*. Springer. 2018, pp. 106–114 (cit. on p. 11).

[Pey10] Gabriel Peyré, Mickael Péchaud, Renaud Keriven, Laurent D Cohen, et al. "Geodesic Methods in Computer Vision and Graphics". *Foundations and Trends extregistered in Computer Graphics and Vision* 5.3–4 (2010), pp. 197–397 (cit. on pp. 5, 17).

[Pha21] Duc Duy Pham, Gurbandurdy Dovletov, and Josef Pauli. "A Differentiable Convolutional Distance Transform Layer for Improved Image Segmentation". *Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020, Tübingen, Germany, September 28–October 1, 2020, Proceedings 42*. Springer. 2021, pp. 432–444 (cit. on pp. 47, 96).

[Pop18] Mihaela Pop, Maxime Sermesant, Pierre-Marc Jodoin, Alain Lalande, Xiahai Zhuang, Guang Yang, et al. *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges: 8th International Workshop, STACOM 2017, Held in Conjunction With MICCAI 2017, Quebec City, Canada, September 10-14, 2017, Revised Selected Papers*. Vol. 10663. Springer, 2018 (cit. on p. 79).

[Rat09] Nathan Ratliff, Matt Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. "CHOMP: Gradient Optimization Techniques for Efficient Motion Planning". *2009 IEEE international conference on robotics and automation*. IEEE. 2009, pp. 489–494 (cit. on p. 111).

[Raz18] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. "Deep Learning for Medical Image Processing: Overview, Challenges and the Future". *Classification in BioApps: Automation of Decision Making* (2018), pp. 323–350 (cit. on p. 13).

[Ren16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-Cnn: Towards Real-Time Object Detection With Region Proposal Networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2016), pp. 1137–1149 (cit. on p. 74).

[Ren13] Xiaofeng Ren and Deva Ramanan. "Histograms of Sparse Codes for Object Detection". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 3246–3253 (cit. on p. 53).

[Ron15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241 (cit. on pp. 3, 6, 15, 18, 74, 79, 85, 86, 116).

[Rup16] Christian Rupprecht, Elizabeth Huaroc, Maximilian Baust, and Nassir Navab. "Deep Active Contours". *arXiv Preprint arXiv:1607.05074* (2016) (cit. on pp. xv, 79–81).

[Rus15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, et al. "Imagenet Large Scale Visual Recognition Challenge". *International Journal of Computer Vision* 115 (2015), pp. 211–252 (cit. on p. 74).

[Sán11] Jorge Sánchez and Florent Perronnin. "High-Dimensional Signature Compression for Large-Scale Image Classification". *CVPR 2011*. IEEE. 2011, pp. 1665–1672 (cit. on p. 53).

[Sap95] Guillermo Sapiro, Ron Kimmel, and Vicent Caselles. "Object Detection and Measurements in Medical Images via Geodesic Deformable Contours". *Vision Geometry IV*. Vol. 2573. SPIE. 1995, pp. 366–378 (cit. on pp. 5, 17).

[Sar22] DR Sarvamangala and Raghavendra V Kulkarni. "Convolutional Neural Networks in Medical Image Understanding: A Survey". *Evolutionary Intelligence* 15.1 (2022), pp. 1–22 (cit. on p. 12).

[Sca22] Christopher Scarvelis and Justin Solomon. "Riemannian Metric Learning via Optimal Transport". *arXiv Preprint arXiv:2205.09244* (2022) (cit. on pp. 5, 17, 112).

[Sch15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A Unified Embedding for Face Recognition and Clustering". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823 (cit. on p. 74).

[Sel17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-Cam: Visual Explanations From Deep Networks via Gradient-Based Localization". *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626 (cit. on p. 11).

[Set96] James A Sethian. "A Fast Marching Level Set Method for Monotonically Advancing Fronts." *Proceedings of the National Academy of Sciences* 93.4 (1996), pp. 1591–1595 (cit. on pp. 2, 6, 17, 41, 114, 130).

[Sim14] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". *arXiv Preprint arXiv:1409.1556* (2014) (cit. on p. 71).

[Sta04] JJ Staal, MD Abramoff, M Niemeijer, MA Viergever, and B Van Ginneken. "Digital Retinal Image for Vessel Extraction (DRIVE) Database". *Image Sciences Institute, University Medical Center Utrecht, Utrecht, the Netherlands* (2004) (cit. on p. 143).

[Sun07] Ganesh Sundaramoorthi, Anthony Yezzi, and Andrea C Mennucci. "Sobolev Active Contours". *International Journal of Computer Vision* 73 (2007), pp. 345–366 (cit. on p. 79).

[Sze15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, et al. "Going Deeper With Convolutions". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (cit. on p. 71).

[Tai14] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. "Deepface: Closing the Gap to Human-Level Performance in Face Verification". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1701–1708 (cit. on p. 74).

[Tsi95] John N Tsitsiklis. "Efficient Algorithms for Globally Optimal Trajectories". *IEEE Transactions on Automatic Control* 40.9 (1995), pp. 1528–1538 (cit. on p. 41).

[Var67] Sathamangalam R Srinivasa Varadhan. "On the Behavior of the Fundamental Solution of the Heat Equation With Variable Coefficients". *Communications on Pure and Applied Mathematics* 20.2 (1967), pp. 431–455 (cit. on pp. 127, 130, 132).

[Vas17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, et al. "Attention Is All You Need". *Advances in Neural Information Processing Systems* 30 (2017) (cit. on pp. 70, 73).

[Ves02] Luminita A Vese and Tony F Chan. "A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model". *International Journal of Computer Vision* 50 (2002), pp. 271–293 (cit. on p. 22).

[Wai13] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. "Phoneme Recognition Using Time-Delay Neural Networks". *Backpropagation*. Psychology Press, 2013, pp. 35–61 (cit. on p. 70).

[Wan18] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, et al. "DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.7 (2018), pp. 1559–1572 (cit. on p. 79).

[Wan22] Risheng Wang, Tao Lei, Ruixia Cui, Bingtao Zhang, Hongying Meng, and Asoke K Nandi. "Medical Image Segmentation Using Deep Learning: A Survey". *IET Image Processing* 16.5 (2022), pp. 1243–1267 (cit. on p. 12).

[Wu13] Guorong Wu, Minjeong Kim, Qian Wang, Yaozong Gao, Shu Liao, and Dinggang Shen. "Unsupervised Deep Feature Learning for Deformable Registration of MR Brain Images". *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013: 16th International Conference, Nagoya, Japan, September 22-26, 2013, Proceedings, Part II 16*. Springer. 2013, pp. 649–656 (cit. on p. 74).

[Wu20] Xiangqiong Wu, Guanghua Tan, Kenli Li, Shengli Li, Huaxuan Wen, Xianyi Zhu, et al. "Deep Parametric Active Contour Model for Neurofibromatosis Segmentation". *Future Generation Computer Systems* 112 (2020), pp. 58–66 (cit. on p. 82).

[Xu98] Chenyang Xu and Jerry L Prince. "Snakes, Shapes, and Gradient Vector Flow". *IEEE Transactions on Image Processing* 7.3 (1998), pp. 359–369 (cit. on p. 31).

[Yan16] Fang Yang and Laurent D Cohen. "Geodesic Distance and Curves Through Isotropic and Anisotropic Heat Equations on Images and Surfaces". *Journal of Mathematical Imaging and Vision* 55 (2016), pp. 210–228 (cit. on pp. 5, 17, 127, 130).

[Zei14] Matthew D Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 818–833 (cit. on pp. xiv, 72).

[Zha16] Jiong Zhang, Behdad Dashtbozorg, Erik Bekkers, Josien PW Pluim, Remco Duits, and Bart M ter Haar Romeny. "Robust Retinal Vessel Segmentation via Locally Adaptive Derivative Frames in Orientation Scores". *IEEE Transactions on Medical Imaging* 35.12 (2016), pp. 2631–2644 (cit. on p. 143).

[Zha20] Mo Zhang, Bin Dong, and Quanzheng Li. "Deep Active Contour Network for Medical Image Segmentation". *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*. Springer. 2020, pp. 321–331 (cit. on p. 83).

[Zha05] Hongkai Zhao. "A Fast Sweeping Method for Eikonal Equations". *Mathematics of Computation* 74.250 (2005), pp. 603–627 (cit. on p. 41).

[Zha15] Liya Zhao and Kebin Jia. "Deep Adaptive Log-Demons: Diffeomorphic Image Registration With Very Large Deformations". *Computational and Mathematical Methods in Medicine* 2015.1 (2015), p. 836202 (cit. on p. 74).

[Zho21] S Kevin Zhou, Hayit Greenspan, Christos Davatzikos, James S Duncan, Bram Van Ginneken, Anant Madabhushi, et al. "A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies With Progress Highlights, and Future Promises". *Proceedings of the IEEE* 109.5 (2021), pp. 820–838 (cit. on p. 112).

[Zhu17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232 (cit. on p. 74).

[Zhu96] Song Chun Zhu and Alan Yuille. "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.9 (1996), pp. 884–900 (cit. on p. 22).

RÉSUMÉ

La segmentation des images médicales est une tâche critique dans la pratique clinique, nécessitant des méthodes précises et fiables pour aider au diagnostic et à la planification du traitement. Cependant, les approches d'apprentissage profond existantes manquent souvent d'interprétabilité et de robustesse, ce qui limite leur application dans des environnements cliniques sensibles. Cette thèse aborde ces défis en proposant deux nouveaux modèles d'apprentissage profond qui intègrent des techniques classiques de traitement d'images pour améliorer la performance et la fiabilité de la segmentation.

La première contribution, le Chan-Vese Attention U-Net, incorpore un mécanisme d'attention basé sur la minimisation de l'énergie de Chan-Vese dans l'architecture U-Net. Cette approche exploite les contraintes géométriques pour guider le processus de segmentation, ce qui permet au modèle de produire des résultats plus précis et plus faciles à interpréter en se concentrant sur les régions pertinentes de l'image et en minimisant les détails non pertinents. La seconde contribution, le Fast Marching Energy CNN, combine les réseaux neuronaux avec le calcul de la distance géodésique pour apprendre les métriques riemanniennes isotropes directement à partir des données, ce qui permet de générer des masques de segmentation robustes qui conservent à la fois les propriétés géométriques et topologiques. Ces méthodes utilisent des transformées de distance différentiables et l'algorithme de marche sous-gradient pour les intégrer dans un cadre différentiables.

En intégrant les techniques traditionnelles de minimisation de l'énergie aux modèles modernes d'apprentissage profond, cette recherche fait progresser le domaine de l'analyse d'images médicales, en offrant des outils plus fiables et interprétables pour la segmentation automatisée. Les résultats de cette thèse ont le potentiel d'améliorer les processus de prise de décision clinique et l'adoption de solutions pilotées par l'IA dans les soins de santé.

MOTS CLÉS

Apprentissage Profond, Vision par Ordinateur, Méchanisme d'Attention, Données Médicales, Contours Actifs, Distances Géodésiques

ABSTRACT

Segmentation of medical images is crucial in clinical practice, requiring accurate and reliable methods to aid diagnosis and treatment planning. However, existing deep learning approaches often need more interpretability and robustness, limiting their application in sensitive clinical environments. This thesis addresses these challenges by proposing two new deep learning models integrating classical image processing techniques to improve segmentation performance and reliability.

The first contribution, the Chan-Vese Attention U-Net, incorporates an attention mechanism based on Chan-Vese energy minimisation into the U-Net architecture. This approach exploits geometric constraints to guide the segmentation process, enabling the model to produce more accurate and easier-to-interpret results by focusing on relevant regions of the image and minimising irrelevant details. The second contribution, Fast Marching Energy CNN, combines neural networks with geodesic distance computation to learn isotropic Riemannian metrics directly from the data, generating robust segmentation masks that preserve geometric and topological properties. These methods integrate differentiable distance transforms and the subgradient walk algorithm into a differentiable framework.

By integrating traditional energy minimisation techniques with modern deep learning models, this research advances the field of medical image analysis, providing more reliable and interpretable tools for automated segmentation. The results of this thesis can potentially improve clinical decision-making processes and the adoption of AI-driven solutions in healthcare.

KEYWORDS

Deep Learning, Computer Vision, Attention Mechanism, Medical Data, Active Contours, Geodesic Distances