Apprentissage profond en traitement d'images : application pour la détection de fumée et feu

Sebastien Frizzi

Université de Toulon
École doctorale 548
LIS - équipe SIIM

Apprentissage profond en traitement d'images : application pour la détection de fumée et feu.

Thèse

SÉBASTIEN FRIZZI

Supervisor: Mr Eric MOREAU
Pofesseur

Toulon, juin 2021

Amazonie 2020

Université de Toulon
École doctorale 548
LIS - équipe SIIM

Apprentissage profond en traitement d'images : application pour la détection de fumée et feu.

Thèse

SÉBASTIEN FRIZZI

Supervisor: Mr Eric MOREAU
Pofesseur

Rapporteurs du manuscrit Septembre, 2021.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Mr Eric MOREAU	Mme Sandra BRINGAY	Mr Salah BOURENNANE
Pofesseur	Professeure	Professeur

Toulon, juin 2021

Université de Toulon
École doctorale 548
LIS - équipe SIIM

Apprentissage profond en traitement d'images : application pour la détection de fumée et feu.

THÈSE dirigée par :

Mr Éric MOREAU Directeur de thèse, Professeur, Université de Toulon
Mr Moez BOUCHOUICHA Co-encadrant, MCF, Université de Toulon

JURY :

Mme Sandra BRINGAY Rapporteur, Professeure, Université Paul Valéry, Montpellier
Mr Salah BOURENNANE Rapporteur, Professeur, École centrale de Marseille, Marseille
Mr Antoine TABBONE Examinateur, Professeur, Université de Lorraine, Vandoeuvre-les- Nancy
Mr Mounir SAYADI Examinateur, Professeur, Université de Tunis- ENSIT, Tunisie

Résumé

Les chercheurs ont établi une forte corrélation entre les étés chauds et l'intensité des incendies de forêts ainsi que leur fréquence. Le réchauffement climatique dû aux gaz à effet de serre tels que le dioxyde de carbone, augmente la température dans certaines parties du monde. Or, les incendies libèrent des quantités importantes de gaz à effet de serre, engendrant une élévation de la température moyenne sur terre induisant à son tour une augmentation des incendies de forêt... Les incendies détruisent des millions d'hectares de zones forestières, des écosystèmes abritant de nombreuses espèces et ont un coût important pour nos sociétés. La prévention et les moyens de maîtrise des incendies doivent être une priorité pour arrêter cette spirale infernale.

Dans ce cadre, la détection de la fumée est primordiale, car elle est le premier indice d'un début d'incendie. Le feu et surtout la fumée sont des objets difficiles à détecter dans les images visibles en raison de leur complexité en termes de forme, de couleur et de texture. Cependant, l'apprentissage profond couplé à la surveillance vidéo peut atteindre cet objectif. L'architecture des réseaux de neurones convolutifs (CNN) est capable de détecter avec une très bonne précision la fumée et le feu dans les images RVB. De plus, ces structures peuvent segmenter la fumée ainsi que le feu en temps réel. La richesse de la base de données d'apprentissage des réseaux profonds est un élément très important permettant une bonne généralisation.

Ce manuscrit présente différentes architectures profondes basées sur des réseaux convolutifs permettant de détecter et localiser la fumée et le feu dans les images vidéo dans le domaine du visible.

Mots clés

Apprentissage profond, CNN, détection du feu et de la fumée, segmentation sémantique, base de données.

Researchers have found a strong correlation between hot summers and the frequency and intensity of forest fires. Global warming due to greenhouse gases such as carbon dioxide is increasing the temperature in some parts of the world. Fires release large amounts of greenhouse gases, causing an increase in the earth's average temperature, which in turn causes an increase in forest fires... Fires destroy millions of hectares of forest areas, ecosystems sheltering numerous species and have a significant cost for our societies. The prevention and control of fires must be a priority to stop this infernal spiral.

In this context, smoke detection is very important because it is the first clue of an incipient fire. Fire and especially smoke are difficult objects to detect in visible images due to their complexity in terms of shape, color and texture. However, deep learning coupled with video surveillance can achieve this goal. Convolutional neural network (CNN) architecture is able to detect smoke and fire in RGB images with very good accuracy. Moreover, these structures can segment smoke as well as fire in real time. The richness of the deep network learning database is a very important element allowing a good generalization.

This manuscript presents different deep architectures based on convolutional networks to detect and localize smoke and fire in video images in the visible domain.

Mots clés

Deep learning, CNN, smoke and fire detection, semantic segmentation, database

Aux générations futures...

Remerciement

Cette thèse, comme la plupart des travaux de recherche, est le fruit d'un travail nécessitant le concours de nombreuses personnes,que je tiens à remercier. En premier lieu, mes remerciements s'adressent à mes responsables de thèse, messieurs Éric Moreau et Moez Bouchouicha. J'ai apprécié la liberté qui m'a été laissée de travailler sur un sujet qui me tenait à coeur.

Je remercie tout particulièrement les collègues qui m'ont apporté du soutien et/ou une relecture des articles déposés durant cette thèse et plus particulièrement à Jean-Marc et Pascal pour son aide en langue anglaise.

Mes remerciements vont également à madame Sandra Bringay, monsieur Salah Bourennane, monsieur Antoine Tabbone et monsieur Mounir Sayadi pour avoir accepté de participer à ce jury de thèse. Je sais, ô combien le temps est précieux et difficile à trouver.

Je tiens à remercier tous les membres de l'équipe SIMM, pour son aide administrative et matérielle et plus particulièrement Nadège et Adoracio. Je remercie également l'IUT par son CARTT et le service de formation de l'université pour son aide matérielle et formative. Un grand merci à l'ensemble des chercheurs et doctorants de l'ENSIT de l'université de Tunis pour leur aide.

Enfin, mes remerciements vont à Charlotte, Lou, Eliott, Maïna, Kylian et Laura pour avoir supporté mon manque de disponibilité durant ces trois années.

Toulon, juin 2021

Sébastien FRIZZI

Table des matières

Table des figures

Liste des tableaux

Chapitre 1: Introduction

Contents

1.1 Les feux de forêt

Les incendies affectent profondément la surface de la Terre et son atmosphère. Ils déversent chaque année des quantités phénoménales de gaz à effet de serre, dont le dioxyde de carbone. Ce gaz anthropique joue un rôle majeur dans l'effet de serre provoquant une augmentation rapide et incontrôlée de la température moyenne sur Terre. Les feux participent à cette augmentation des gaz à effet de serre. De plus, les incendies détruisent des végétaux essentiels au stockage du dioxyde de carbone dégagé par les activités humaines. Un cercle vicieux est engagé et mènera, selon certaines études, au doublement des quantités de dioxyde de carbone expulsé dans l'atmosphère par les feux de forêts d'ici la fin du siècle. Même si les feux de forêt font partie de notre histoire, des analyses sédimentaires sur des charbons de bois[14] et isotopiques dans les calottes glaciaires [15] indiquent que la quantité de biomasses brulées dans les deux millénaires passés est moins importante que celle consommée depuis les deux derniers siècle.

L'Organisation des Nations Unies pour l'Alimentation et l'Agriculture (FAO Food and agriculture organisation) pointe du doigt une diminution importante et constante des surfaces forestières pour le continent Africain et Sud américain *Figure 1.1*. Cette réduction de surface des forêts peut être due à la déforestation pour la mise en culture et/ou aux feux qui dans la majeure partie des cas est imputable aux activités anthropiques.

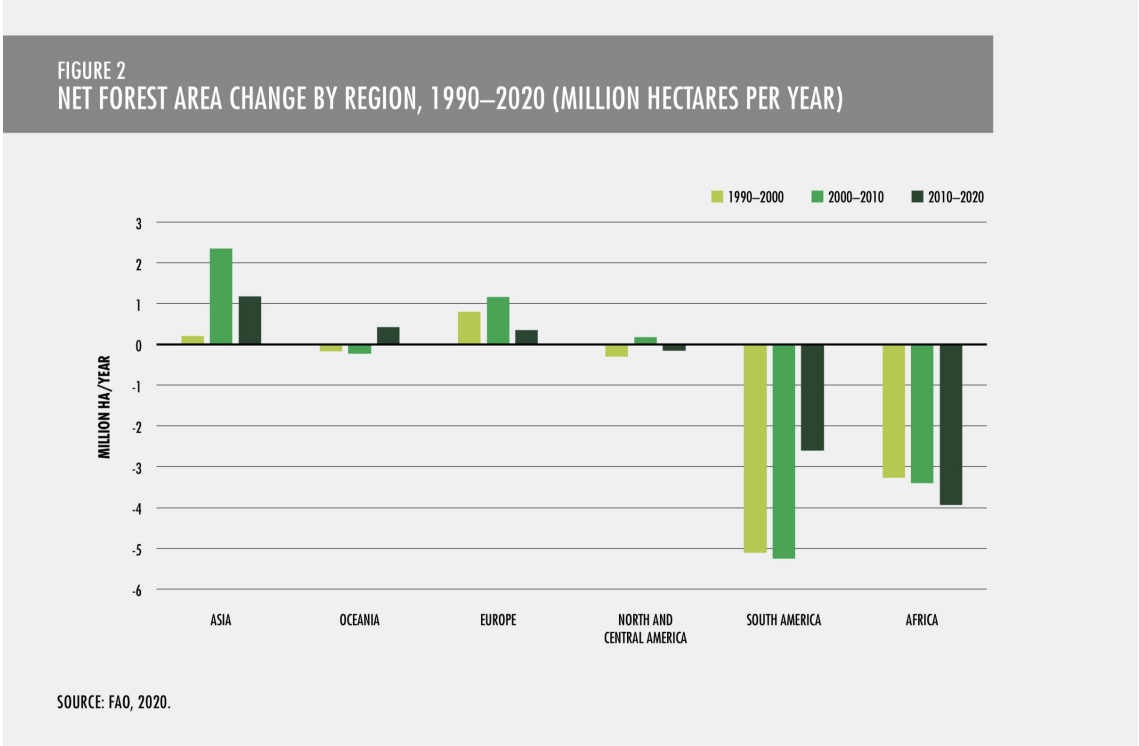


FIGURE 1.1 – *Variation de la surface nette des forêts sur les trois dernières décennies (1990 - 2020)*

En avril 2020, le nombre de feux sur l'ensemble du globe a augmenté de 13% par rapport à l'année 2019 [16]. La responsabilité humaine de ces feux est estimée à 75%. Au-delà de l'impact négatif sur l'effet de serre engendrant une augmentation de la température globale de la terre joue un rôle extrêmement négatif sur la biodiversité tant animale que végétale ainsi que sur le dérèglement des écosystèmes. Le changement climatique expose la végétation à de fortes pressions environnementales et réduit le temps de régénération des écosystèmes. Ce déséquilibre favorise la diffusion des maladies et le développement d'espèces invasives.

En Europe, 40 mille feux sont rapportés chaque année. Entre 2010 et 2017, en moyenne 350 mille hectares ont été affectés par les incendies chaque année sur les 180 millions d'hectares de forêt que compte l'Union européenne [17].

Entre 2000 et 2017 [18] et [19] :

- 8,5 millions d'hectares ont été atteints par un feu de forêt.
- 611 personnes ont perdu la vie par cause directe d'un incendie.
- 54 milliards de perte directe sur l'économie.

La Commission européenne encourage la recherche et l'innovation afin de réduire et d'anticiper la gestion des feux de forêt. Une récente étude scientifique commandée par l'Union européenne a contribué à identifier les principaux défis liés au management et gestion des feux de forêt sur le continent européen *figure 1.2*.

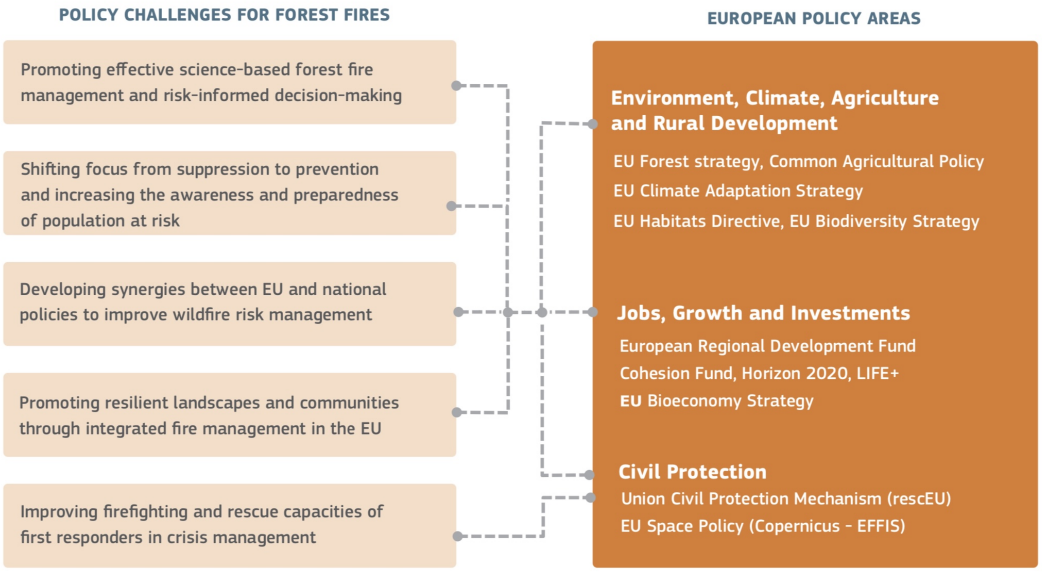


FIGURE 1.2 – *Politique européenne face aux risques d'incendie - 2018)*

Le premier objectif souligne la promotion des sciences en relation avec le management et la gestion des risques d'incendie. Notre projet entre parfaitement dans cet axe. La surveillance en temps réel des départs de feu est un enjeu majeur et apporterait une aide précieuse aux pompiers. Les images aériennes offrent aux forces d'intervention au sol les zones dangereuses par la surimposition des zones de feu et de fumées sur des cartes. Ces informations pourraient réduire le temps d'intervention sur les zones touchées.

La France et plus particulièrement le sud de la France est frappée par les feux de forêt. Le climat méditerranéen chaud et sec de l'été combiné à des vents violents favorise les départs de feux. Ces dernières années, des surfaces importantes ont subi l'assaut des flammes. Au-delà de l'impact sur la biodiversité et les biens des habitants, le bilan humain, la plupart du temps chez les pompiers et la sécurité civile, est trop important. Depuis 1973, l'état s'est doté, pour la zone méditerranéenne, très sensible aux feux de forêt, d'une base de données statistiques nommée Prométhée [20]. Cette base étatique et fiable recense les incendies temporellement et spatialement. Les services SDIS, DDTM, ONF, gendarmerie et police alimentent cette base. L'interrogation de cette base fournit la date, le lieu, le type de feux et la surface en hectares brulés. Chaque année, une carte de synthèse du pourtour méditerranéen par département est établie permettant à la sécurité civile de s'organiser les mois d'été.

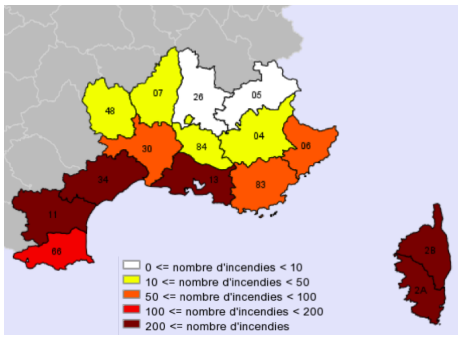


FIGURE 1.3 – *Carte du nombre de feux par département pour l'année 2016. Pourtour méditerranéen.*

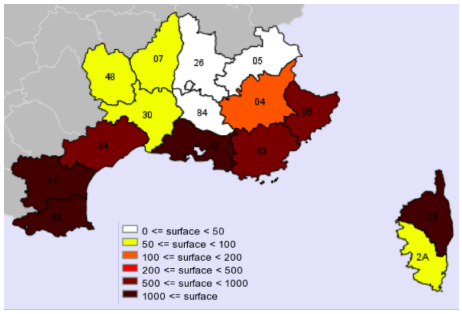


FIGURE 1.4 – *Carte de la surface en hectares brulés par département pour l'année 2016. Pourtour méditerranéen.*

Les incendies de forêt ont des répercussions sur l'économie, l'écologie et la sécurité des territoires dans le monde. Avec le réchauffement climatique, ce phénomène risque de fortement s'amplifier dans les prochaines décades. Les politiques dans leur grande majorité sont conscients des risques et des pertes économiques dues à ce phénomène. Des solutions de surveillance et de management via l'apprentissage profond et le traitement des images peuvent fournir des solutions rapides et pertinentes.

1.2 Structure de la thèse

Cette thèse s'est déroulée au sein du LIS (Laboratoire d'Informatique et des Systèmes) et plus particulièrement dans le pôle SIIM (SIgnal et IMages) de l'université de Toulon. La problématique abordée était de trouver des architectures de réseaux de neurones permettant de détecter et localiser le feu et la fumée dans des images du spectre visible issues de caméras. Cette détection compatible avec le temps réel a pour objectif d'apporter une aide effective aux pompiers dans le domaine de la prévention du feu ou lors des phases de lutte contre celui-ci. L'apprentissage profond et plus particulièrement l'utilisation des réseaux convolutifs s'est imposée rapidement pour deux raisons essentielles. La première réside dans le fait que les réseaux convolutifs sont calqués sur la perception humaine et le fonctionnement du cortex visuel des mammifères. Ce type de réseau est bien adapté à la détection ou à la segmentation d'objets. La structure spatiale est conservée dans les cartes de caractéristiques successives. Dans le cadre d'un apprentissage supervisé, il ne demande pas la présence d'un expert du domaine pour déterminer le vecteur de caractéristiques. Le réseau détermine par la phase d'entraînement les caractéristiques des objets à détecter ou segmenter. Enfin, l'évolution des processeurs, plus particulièrement des cartes graphiques GPU permet dorénavant d'entraîner ce type de réseau en quelques jours voire quelques heures. Il est possible également d'utiliser ces structures de réseau lourdes dans des temps compatibles avec le temps réel sur des ordinateurs classiques ou sur des smart phones. La prévention est un levier important dans la lutte contre les incendies. Plus vite un feu est détecté, plus vite il sera maîtrisé. La fumée reste le premier indice d'un départ de feu et doit être détectée et localisée afin d'envoyer des moyens d'interventions. La difficulté de détection réside dans la variabilité des paramètres de la fumée (luminosité, couleur, forme

...). Les réseaux convolutifs sont capables de détecter et segmenter les zones de fumée et/ou de feu permettant d'informer les pompiers sur l'étendue, la direction de propagation du feu ; apportant une aide précieuse aux forces d'intervention au sol et dans les airs. La détection des zones de feu en temps réel serait un atout pour le centre de commande d'intervention. L'information serait envoyée d'un drone ou un hélicoptère vers une base au sol pour le traitement des informations en temps réel. Une surimposition des routes d'accès et des habitations via un logiciel de SIG pourrait être envisagée. Elle permettrait de donner des informations sur les routes praticables par les pompiers lors de l'intervention.

1.3 Les contributions

Les contributions relatives à la détection du feu et de la fumée apportées à la communauté scientifique ont été les suivantes :

- Une architecture légère basée sur un réseau convolutif permettant une détection du feu et de la fumée sur des images RGB dans des temps réels [21]. Cette conception de structure s'est accompagnée d'une base de données d'images de feux et de fumées. Nous avons montré que la détection du feu et de la fumée pouvait être réalisée à partir d'un seul réseau convolutif avec une très bonne précision ;
- Une structure de réseau de segmentation appliquée à la segmentation du feu et de la fumée sur des images vidéos dans le spectre du visible [22]. Réseau qui, pour des images de faible définition, prédit et détermine la classe des pixels de l'image dans des temps compatibles avec le temps réel. Cette étude s'est accompagnée de la réalisation d'une base de données composée des images RGB ainsi que des masques de feu et de fumée ;
- La comparaison de performance d'un réseau entraîné sur deux bases de données différentes [23].

J'ai également contribué lors de mon doctorat aux articles de recherches suivants :

- Un article synthèse sur la détection du feu et de la fumée dans le domaine du visible et de l'infrarouge [24] ;
- Deux articles sur la détection d'hernies discales lombaires à partir d'images médicales à l'aide d'un réseau convolutif. Les disques et apophyses sont segmentés automatiquement permettant à terme une classification du type de hernie discale [5] , [25]. Ces articles ne sont pas en rapport avec mon sujet de thèse. Toutefois, ils sont basés sur la même structure de réseau que celle utilisée pour la segmentation du feu et de la fumée avec des bases d'apprentissage et de test différentes.

La première partie de ce mémoire est consacrée à l'état de l'art relatif à l'apprentissage profond. Nous commencerons par réaliser un tour d'horizon historique des découvertes tant biologiques que technologiques du fonctionnement des neurones. Nous nous plonge-

rons ensuite dans la détection du feu et de la fumée avec des méthodes dites "classiques" n'utilisant pas l'apprentissage profond. Ensuite, nous parlerons de la particularité du réseau convolutif et de son évolution. Nous poursuivrons cette partie en explorant le domaine de l'apprentissage profond dédié à la segmentation. Pour finir, nous définirons les points importants nécessaires à l'entraînement des réseaux profonds.

La seconde partie sera dédiée à la détection du feu et de la fumée par réseaux convolutifs.

La troisième partie traitera de segmentation du feu et de la fumée à partir d'image du visible.

La dernière partie conclura sur les apports de cette thèse et ouvrira des portes sur de potentielles perspectives à envisager.

Chapitre 2: État de l'art

Contents

2.1 Histoire de l'apprentissage artificiel

L'apprentissage profond est de nos jours très populaire et employé abondamment dans la vie courante. Cet état de fait n'a pas toujours été le cas. La découverte des réseaux de neurones artificiels a fortement été inspirée par l'anatomie biologique animale et humaine. Beaucoup de scientifiques et philosophes ont œuvré pour percer les mystères de la pensée et du transport de l'influx nerveux. Cette longue histoire a été ralentie par la religion et l'éthique des diverses époques. Un bref rappel historique semble nécessaire pour comprendre les réseaux de neurones actuels et rendre ainsi hommage à tous ces scientifiques, la plupart du temps inconnus ou peu connus, qui ont apporté leur pierre à l'édifice des réseaux de neurones artificiels menant à l'apprentissage profond.

2.1.1 Antiquité et moyen Âge

Les premières traces de la volonté de modéliser la pensée humaine remontent aux philosophes de la Grèce antique. Aristote *(384-322 av. J.-C.)* dans son essai "motu animalium" datant de 350 av. J.-C.[26] réalise des hypothèses sur la manière dont l'âme commande le corps des créatures vivantes. Contrairement à notre définition actuelle du le cerveau constituant le centre de la conscience et le carrefour de toutes les connections nerveuses, Aristote positionne le centre des sens dans le coeur. L'animal se meut du fait de la coopération du désir et d'une pensée ou représentation comme une sensation ou une image. Toutes ces informations sensitives sont transmises vers le coeur qui permet de produire en retour un mouvement volontaire ou involontaire des organes internes ou au déplacement de l'animal.

Il n'y a pas de place pour le cerveau dans la chaine causale de sensation/action des animaux. Le cerveau était pour Aristote une "machine thermique" refroidissant le sang échauffé par les émotions. La vision d'Hippocrate *(460-379 av. J.-C.)* s'opposa à la croyance millénaire défendue par Aristote selon laquelle la localisation de l'activité mentale se situe dans le coeur. Pour lui, le centre de la pensée se trouvait dans le cerveau qui gouverne les sentiments et émotions.

Cette idéologie classique du cœur au centre de l'activité mentale resta fortement ancrée dans les inconscients dû certainement au ressenti des émotions comme les frissons en cas de peur soudaine et de chaleur dans des phases d'énervement. Des expressions de cette croyance résident encore dans notre langage quotidien comme : *"Briser le coeur",es "Apprendre par coeur", "Avoir un coeur de pierre", etc.* Le médecin grec Claude Galien *129-216 apr. J.-C.* fut un spécialiste de la dissection des animaux qui pratiqua ses expérimentations cachées à cause des croyances très fortes interdisant ce type de pratique. Il confirma l'hypothèse d'Hippocrate situant le centre de la pensée et de l'âme dans le cerveau et non plus dans le cœur. Il décrivit avec justesse le parcours de l'influx nerveux avec quelques erreurs qui se répercutèrent pendant des siècles, à cause de ses dissections sur les animaux et le transfert sur l'homme. Galien décomposa le cerveau en deux grandes parties : l'encéphale siège des sensations et le cervelet semblant commander les muscles. Les nerfs n'étaient, pour lui, que des tubes vides dans lesquels circulaient des fluides. Une contribution philosophique majeure de Galien fut de mettre en forme le concept selon lequel les objectifs de Dieu sont explicables par l'observation de la nature. L'église entérine la doctrine de Galien confortant l'existence d'un Dieu tout puissant et unique créateur de l'Homme. Cette doctrine perdurera des siècles sous peine de représailles de la part de l'église.

FIGURE 2.1 – *Vivisection sur un porc par Galien*

Au $XIII^{e}$ siècle la première théorie de dualité corps et esprit/âme voit officiellement le jour dans l'ouvrage *somme théologique* de Thomas D'Aquin *(1224-1274)* qui est un texte fondateur de la doctrine chrétienne. Idée selon laquelle le corps et l'esprit appartiennent à des domaines différents. L'âme ou l'esprit humain est constitué d'une substance immortelle échappant aux lois de la physique.

Toute la période du moyen Âge reste la période des interdits et des tabous pour les avancées scientifiques et particulièrement médicales. L'homme en tant que création de Dieu

est intouchable. La théorie de Galien se maintient dans la peur.

2.1.2 Renaissance - Philosophie de l'esprit

Il faut attendre la renaissance pour que la connaissance du cerveau et du système nerveux humain se précise. À cette époque, Léonard de Vinci *(1452-1519)* réalise des dissections humaines et produit de nombreux croquis anatomiques de qualité. André Vésale *(1515-1564)* poursuit le travail de Léonard de Vinci en réalisant des croquis anatomiques très détaillés. Il décrivit très précisément l'anatomie du cerveau et distingua la substance grise et blanche.

René Descartes au milieu du 17ème siècle soumet dans ses hypothèses un nouveau dualisme entre le corps matériel et l'âme/esprit immatériel. L'esprit ou l'âme et le corps appartiennent à deux univers différents. Il explique le fonctionnement du corps uniquement d'un point de vue mécanique. Il compare le corps humain et le cerveau à un orgue d'église. Pour lui, des soufflets insufflent le mouvement de fluides vers une série de canalisations et de réservoirs. Le fonctionnement du corps humain n'a nul besoin de l'âme ou de l'esprit pour opérer, toutefois l'esprit peut ordonner au corps d'agir. Descartes explique le principe de la vision d'un objet par des phénomènes hydrauliques. La question qui tarauda Descartes était d'expliquer dans sa théorie, comment un élément immatériel comme l'esprit peut-il influencer un élément matériel comme le corps. Il émit l'hypothèse que les êtres animés interagissaient avec leur esprit par le cerveau. En réalité, la connexion entre l'esprit/âme et le corps était la glande pinéale qui se situe au centre du cerveau, entre les deux hémisphères. L'image d'un objet se projette sur la rétine qui commande une série de tuyaux particuliers représentant la scène visuelle. La glande pinéale force "l'esprit animal" à s'écouler vers les muscles nécessitant le mouvement désiré. La mémoire consisterait à un renforcement sélectif de certains tuyaux au détriment d'autres. Cette vision de l'apprentissage préfigurait l'idée de renforcement des liaisons neuronales du cerveau lors de l'apprentissage. L'esprit commande le corps qui peut exister en dehors de celui-ci. Ce dualisme continua d'apparaître de manière privilégiée dans la philosophie d'une grande partie des civilisations.

Le médecin anglais Thomas Willis va placer le cerveau au centre du comportement animal et humain. Il réalise des dissections sur des cadavres de criminels exécutés ou de ses patients avec leur accord ante mortem. Ses études le mènent à définir le terme de *neurolgia* ou neurologie. Il réalise des comparaisons du système cérébral entre l'Homme et les animaux. Il notera des cervelets très similaires, malgré une structure du cerveau différente. Dans *cerebri Anatome* il identifie et définit les parties du cerveau ainsi que le parcours des nerfs crâniens. Dans *Pathologiæ Cerebri et Nervosis Generis Specimen* il relie les troubles psychiques à des altérations des parties du cerveau. Avec cette analyse, il indique le rôle primordial du cortex dans le comportement de l'Homme et des animaux.

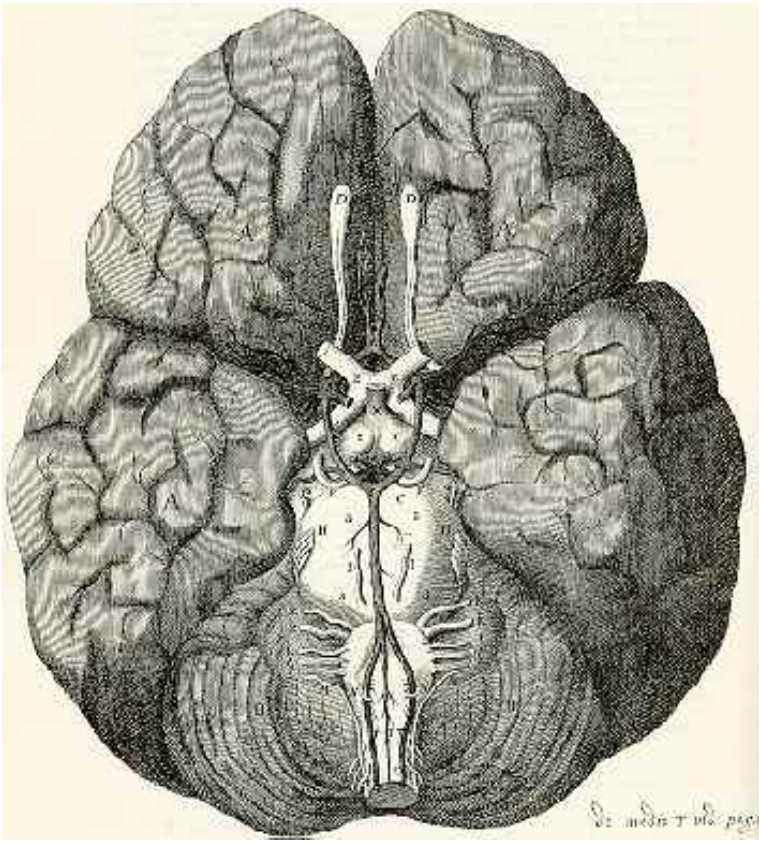


FIGURE 2.2 – *Dessin montrant les artères du cerveau - Celebri anatome - Thomas Willis. 1662*

À la fin du $XVII^e$ siècle le philosophe anglais John Locke réfute l'idée communément admise selon laquelle l'âme et donc l'esprit contiendrait à notre naissance passivement des idées indépendamment de notre expérience. Il écrit dans le Livre premier *des Essais* : *"Supposons donc qu'au commencement l'âme est ce qu'on appelle une table rase, vide de tous les caractères sans aucune idée, quelle qu'elle soit. Comment vient-elle à recevoir des idées ? Par quel moyen en acquiert-elle cette prodigieuse quantité que l'imagination de l'homme, toujours agissante et sans bornes, lui présente avec une variété presque infinie ? D'où puise-t-elle tous ces matériaux qui sont somme le fond de tous ses raisonnements et de toutes ses connaissances ? A cela je réponds en un mot, de l'expérience : c'est le fondement de toutes nos connaissances, et c'est de là qu'elles tirent leur première origine"*

John Locke rejette donc les fondements de la philosophie Cartesienne en maintenant que tout Homme nait sans idée innée et forge son esprit ou son âme de ses expériences. Dans son travail, il introduit la notion "d'association d'idées". L'esprit est organisé par le principe de l'association et que les items qui s'assemblent dans l'expérience de l'individu iront ensemble dans la pensée. La nature des idées, concepts et autres états mentaux sont expliqués en terme " d'états cognitifs" soulignant le processus cognitif incluant connaissances, croyances, compréhension, etc. Pour John Locke, le processus cognitif se résume simplement à des représentations mentales. Ses travaux suggèrent que les associations extrinsèques apparaissent volontairement ou par chance, la force de l'impression de l'idée

peut renforcer l'association. De plus, certains items sont plus facilement liables ensemble que d'autres. On voit émerger le terme d'apprentissage et de renforcement cognitif.

2.1.3 XIX^e Siècle - Biologie du cerveau

Après des dizaines d'années de théories sur le fonctionnement cognitif des Hommes, les prémisses de la théorie de la conception neurale de l'esprit sont proposées au XIX^e siècle par le philosophe anglais Alexander Bain et William James. L'idée centrale de leur théorie repose sur le fait que les pensées et les mouvements du corps reposent sur l'activité des neurones dans le cerveau.

Une nouvelle science voit le jour au début du XIX^e siècle : Phrénologie. Cette science soulèvera de nombreuses controverses. Le médecin allemand Franz Gall *(1758-1828) Livre Anatomie et physiologie du système nerveux en général et du cerveau en particulier F.J Gall, G Spurzheim - 1810* [27] émet l'hypothèse selon laquelle, la morphologie du crâne reflète les facultés mentales et intellectuelles d'un individu. Il réalisa une cartographie corticale composée de 27 régions.

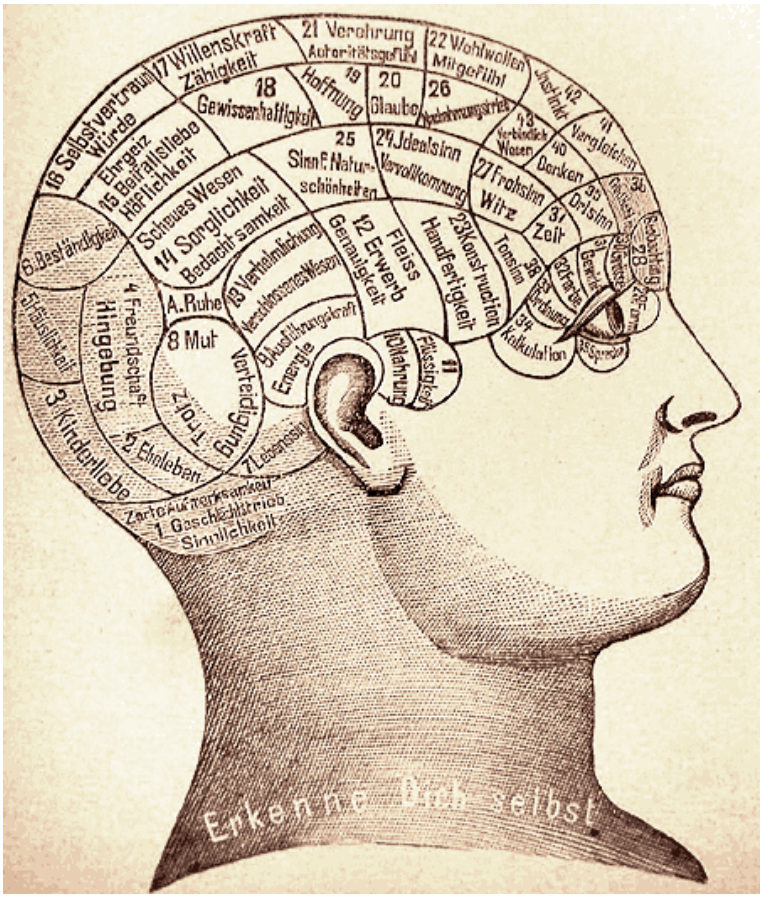


FIGURE 2.3 – *Carte corticale crânienne - Phrénologie - Franz Gall. 1662*

La phrénologie voulant tout expliquer par la forme du crâne partit sur des chemins tous azimuts discréditant cette science. Gall crut pouvoir déterminer une corrélation certaine entre la protubérance des yeux et les capacités cérébrales de ses étudiants. D'autres cher-

cheurs déterminèrent la bosse des mathématiques, etc. Malgré la fantaisie de cette théorie fausse, la phrénologie à ouvert la voie de la cartographie corticale. Le médecin, anatomiste Paul Broca *(1824-1880)* va démontrer grâce à un patient, M Leborgne surnommé "Tan" à cause de la seule syllabe qu'il pouvait prononcer, l'existance de régions du cerveau spécialisées dans la parole. Des études sur les animaux démontrèrent la véracité des hypothèses de Paul Broca selon laquelle le cerveau est composé de zones spécialisées. Malgré ses idéologies racistes et sexistes, expliquant que la petite taille du cerveau des femmes était corrélée à la petite taille de son corps et expliquait son infériorité intellectuelle, Paul Broca est un pionnier en anthropologie physique.

En 1837, le physiologiste, histologiste et cytologiste Théodore Schwann *(1810-1882)* dans son l'ouvrage *Recherches microscopiques sur la similarité de structure et de développement des cellules animales et végétales* [28] affirme l'importance de la cellule en tant qu'entité élémentaire des êtres vivants. L'importance de cette découverte et l'amélioration des microscopes lancent les chercheurs vers l'étude et la visualisation des cellules des êtres vivants. Ceci marque la naissance de l'histologie. Il faudra tout de même attendre encore quelques années pour pouvoir observer correctement les cellules neurales grâce entre autres aux recherches de Franz Nissl *(1860-1919)* et Camillo Golgi *(1843-1926)*. Ce dernier parvient à colorer les neurones et à en déterminer la forme. Toutefois, ils restent sur l'idée que les excroissances du neurone sont constituées de canalisation dans lesquelles s'écoulent des fluides.

Il faudra attendre les travaux de l'histologiste et neuro-scientifique espagnol Santiago Ramon y Cajal *(1852-1934)* pour définir les structures fines du système nerveux et particulièrement celles du cerveau. Il dépasse les hypothèses de structure en forme de canalisation de Camillo Golgi en démontrant la stricte séparation des neurones entre elles.

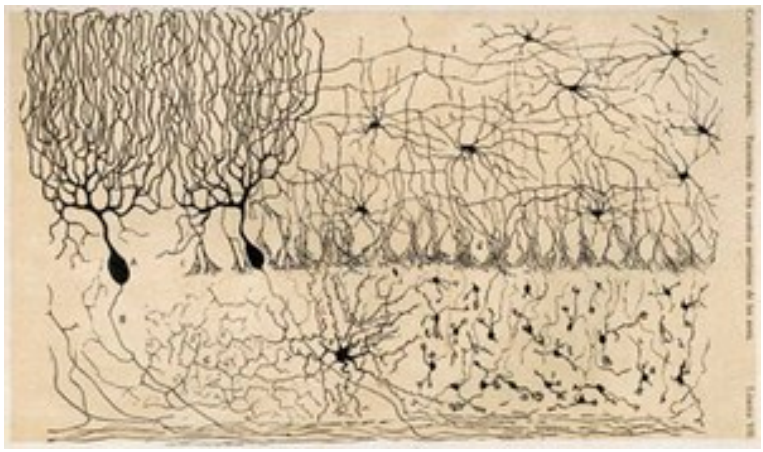


FIGURE 2.4 – *Dessin de la main de Ramon y Cajal de cellules cerebelleuses de poulet, tiré de Estructura de los centros nerviosos de las aves, Madrid, 1905*

Le XIX^e siècle fut celui de l'étude du cerveau et des neurones. Beaucoup d'études furent menées sur l'aspect fonctionnel de la transmission de l'information entre les neurones. Von Helmholtz *(1821-1894)*, le physiologiste et physicien, mesura la vitesse de l'influx nerveux sur une grenouille. Cette impulsion électrique sera nommée potentiel d'action plus

tard. Il faudra attendre 1950 et le développement du microscope électronique pour donner définitivement raison à la théorie de Ramon y Cajal sur l'indépendance des neurones en visualisant l'espace inter synaptique.

Les neurones sont donc indépendants et parcourus par des impulsions électriques.

2.1.4 XX^e siècle - Modélisation du cerveau

L'évolution de la technologie permettra au XIX^e siècle de comprendre le fonctionnement des réseaux de neurones dans le cerveau et système nerveux par la découverte des synapses, des neurotransmetteurs, des potentiels d'actions délivrés par le coeur des neurones.

Neurone biologique

Le néocortex est un tissu organique de 2 à 2,5 millimètres d'épaisseur recouvrant les deux hémisphères du cerveau. Il est composé de cellules appelées neurones. Le néocortex humain est constitué généralement de 6 couches.

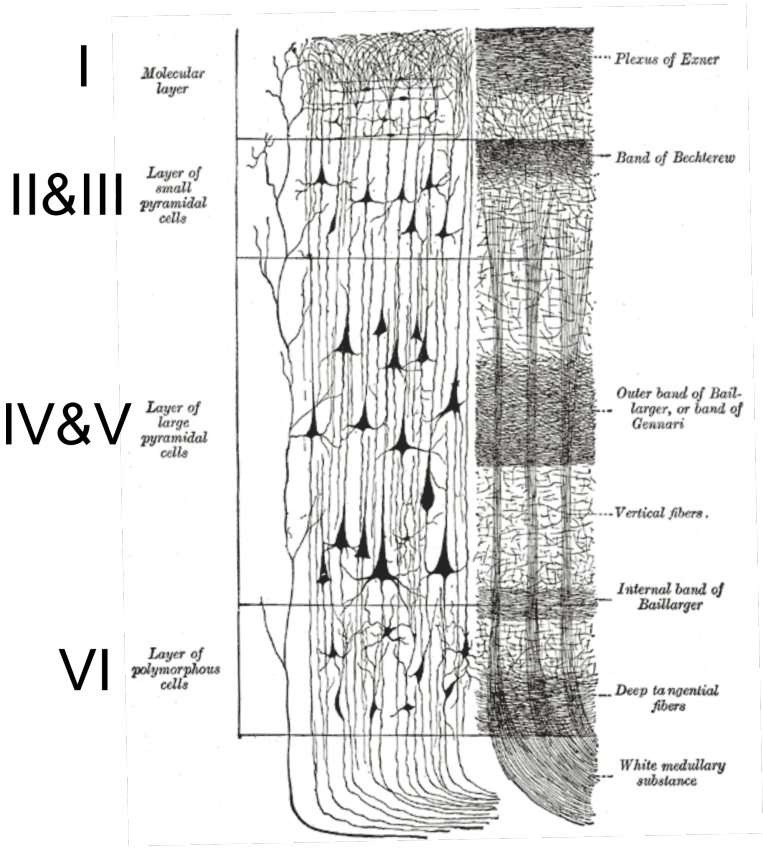


FIGURE 2.5 – *Les 6 Couches du néocortex (Organisation histologique du néocortex. Couches II et III et IV et V ont été regroupées) Henry Gray (1825–1861). Anatomy of the Human Body. 1918.*

Chaque couche possède une population neuronale différente et projette des connexions dans les autres couches. Les couches profondes quant à elles, reçoivent des connexions

provenant de cet arrangement et ne correspond pas à un simple empilement de neurones. Ils s'organisent en unité fonctionnelle et sous forme de colonnes perpendiculaires à la surface du cortex (Fig 4). La brique de base de nos capacités cognitives est le neurone. Giorgio A Ascoli dans son livre *Trees of the brain, Roots of the mind* utilise des métaphores arboricoles pour décrire ceux-ci. Il existe différents types de neurones dans le cerveau (granulaire, pyramidal ...) comme différents types d'essences d'arbres dans une forêt. Chaque arbre possède des caractéristiques propres et une forme propre, les différents types de neurones également. Le neurone possède, comme l'ensemble des cellules de notre corps, un noyau et une membrane cellulaire. Toutefois, leur forme constituée de nombreuses branches ramifiées leur confère une spécialisation dans la réception et l'émission de signaux électrochimiques.

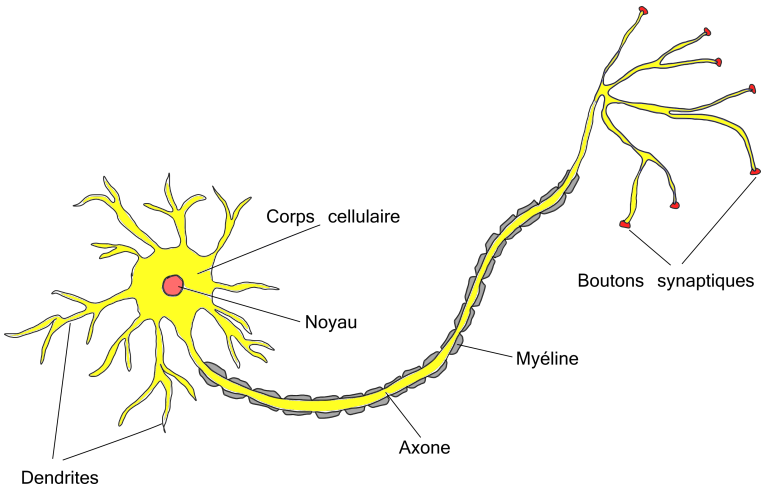


FIGURE 2.6 – *Le neurone avec son corps cellulaire, ses dendrites, axone et synapses.*

La communauté scientifique étudie les neurones depuis près de 200 ans et a montré que ces cellules se sont conservées durant 100 millions d'années d'évolution. Ceci prouve la grande efficacité du processus biologique de communication mis en place par ces cellules.

Le neurone est composé de petites excroissances cellulaires tubulaire de l'ordre de quelques microns nommés dendrites et axones. L'axone se subdivise en son extrémité pour former des terminaisons axonales reliées à la synapse. Les dendrites reçoivent les signaux envoyés par les axones des neurones connectés. Les axones quant à eux, émettent les signaux vers les autres neurones via leurs synapses. De par sa fonction, l'axone est généralement plus long que les dendrites. Les dendrites correspondent aux entrées du neurone et les terminaisons axonales aux sorties. Un neurone possède de 1000 à 10 000 connexions avec d'autres neurones. Le corps cellulaire ou soma correspond au centre de contrôle de la communication. Il reçoit les signaux des autres neurones provenant des dendrites, réalise une somme et transmet l'information vers les terminaisons de l'axone. La communication intra neuronale se réalise par variations de potentiels dits potentiels d'action de l'ordre de quelques millivolts.

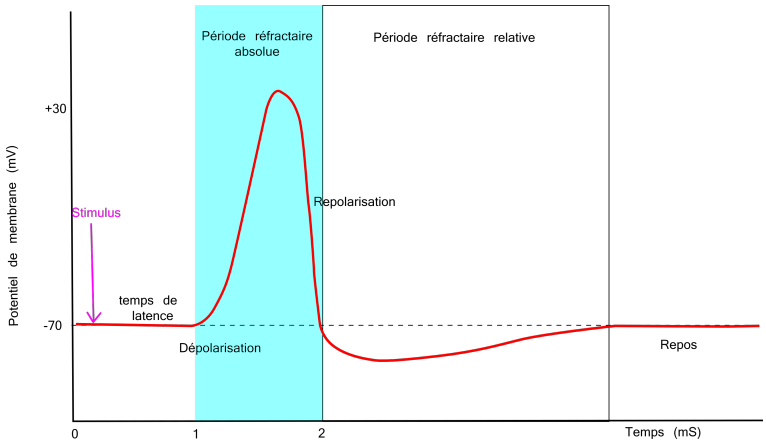


FIGURE 2.7 – *Potentiel d'action du neurone.*

Il s'agit d'une dépolarisation brève et réversible se propageant le long de la membrane cellulaire. L'émission du pic du potentiel d'action dure approximativement d'une milliseconde. Le neurone a besoin d'un temps de repos d'environ une milliseconde, appelée période réfractaire absolue, pour se repolariser avant de pouvoir recevoir de nouveaux signaux. La période réfractaire absolue est suivie d'une période réfractaire relative durant laquelle il est possible de mettre le feu à un nouveau potentiel d'action avec un seuil plus élevé. Avant de retrouver son état de polarisation normal. Le rôle de la soma est très important, elle réalise une sommation spatiale et temporelle des potentiels posts synaptiques reçus et décide si le neurone est actif ou non actif en fonction des signaux reçus par les dendrites. Si la somme des potentiels posts synaptique via les neurotransmetteurs des dendrites est en dessous du seuil d'activation, il n'y aura pas de potentiel d'action. Dès que ce potentiel passe au-dessus du seuil, un potentiel d'action est libéré. Le potentiel d'action transporté par l'axone est d'une amplitude fixe sous forme de pic de potentiel appelé « Spike ». La réaction neuronale s'apparente à un fonctionnement binaire (tout au rien). Le trajet du signal le long des axones ou dendrites peut subir une atténuation ou des perturbations. On peut assimiler la transmission de signaux entre deux neurones à un message Morse du temps du Far West. Le neurone émetteur va envoyer le potentiel d'action ou « Spike » à différents intervalles de temps, correspondant à une information à transmettre. Le trajet est soumis à des perturbations le long de la ligne. Le neurone récepteur joue le rôle d'un opérateur. Il pourra décoder uniquement les "beeps" qu'il entend et ainsi retranscrire le message.

La probabilité de transfert d'un signal entre deux neurones dépend de facteurs propres à chaque neurone. Lorsque le potentiel d'action atteint la synapse, des molécules chimiques appelées neurotransmetteurs sont libérées. Ces neurotransmetteurs partent de la terminaison axonale, traversent la synapse pour rejoindre la dendrite d'un autre neurone.

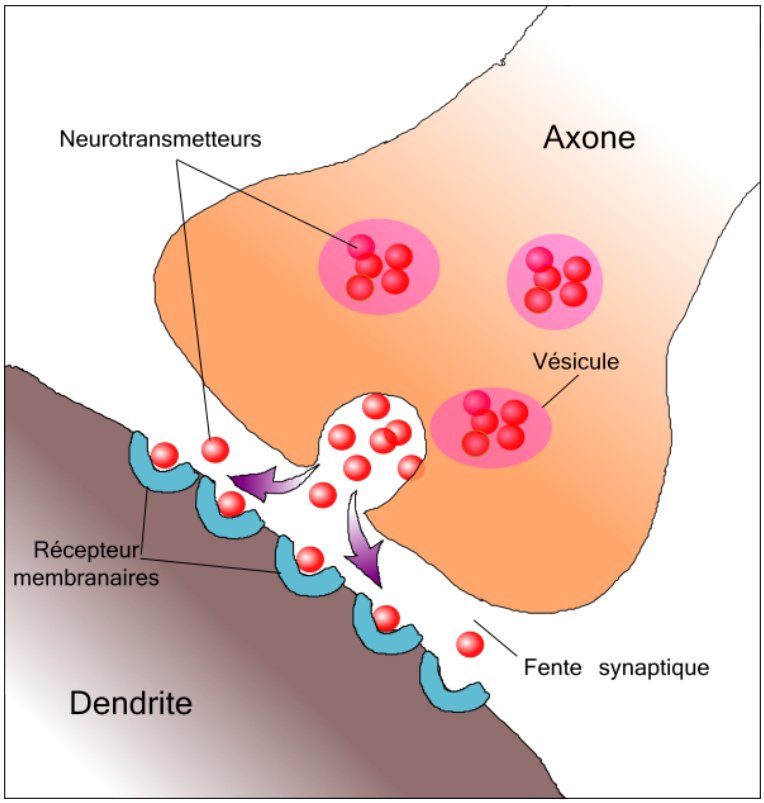


FIGURE 2.8 – *Fente synaptique et transmission des neurotransmetteurs*

Il existe deux types de neurotransmetteurs : les excitateurs et les inhibiteurs. Ce type de transfert de l'information par messager chimique permet de moduler la force du signal entre l'axone et la dendrite. Plus le nombre de neurotransmetteurs excitateurs libéré est important et plus fort sera le signal transmis. La transmission de l'information au niveau d'une synapse peut être modélisée mathématiquement par un poids de transfert synaptique lié à la quantité de neurotransmetteur inhibiteur ou excitateurs libérés. Le poids de transfert de chaque synapse varie dans le temps en fonction du vécu du sujet. Lors de l'apprentissage, le cerveau reçoit des informations du monde extérieur. Ces influx nerveux sont envoyés vers les différentes régions corticales par des groupements de neurones. Chaque neurone va renforcer les liaisons synaptiques si cet apprentissage a déjà été vu ou atténuer les liaisons synaptiques qui sont moins utilisées. Cet apprentissage s'apparente à augmenter ou réduire les poids de transfert synaptique de chaque neurone. Ces processus mettent en évidence la variabilité des connexions neuronales ou plasticité neuronale tout au long de la vie.

Neurone artificiel

En parallèle de la découverte biologique du système nerveux, des scientifiques tentent de modéliser le fonctionnement de celui-ci d'un point de vue électrique ou calculatoire.

L'histoire du neurone artificiel démarre dans les années 1940 avec la publication d'un article de recherche par McCulloch et Pitts [29], inspiré des théories d'Alan Turing. Warren McCulloch est un neurologue et psychiatre passionné de philosophie et de mathématiques.

Il est persuadé que le but de la neurologie et de la psychiatrie est d'expliquer l'esprit en termes de mécanisme neural. Pour cela, il met au point une théorie psychologique basée sur « l'atome mental » appelé psychon. Cet élément insécable de l'état mental permettrait par combinaison d'expliquer le fonctionnement de l'esprit et les pathologies inhérentes. Durant son internat de neurologie à l'hôpital Bellevue de New York, il développe un intérêt au sujet des boucles fermées du système nerveux. Il pense que ces boucles fermées sont responsables de certains désordres mentaux. Entre autres, il tente d'expliquer les tremblements de la maladie de Parkinson par une connexion en boucle entre la moelle épinière et les muscles. En 1929, il lance l'idée selon laquelle, les impulsions axonales (potentiel d'action) "tout ou rien " correspondent aux atomes mentaux. Sa théorie s'oriente peu à peu vers la théorie de l'information échangée entre les neurones. Afin d'étayer sa théorie, il tente d'appliquer l'algèbre booléenne pour expliquer le comportement de l'esprit. En 1934 McCulloch déménage à Yale dans le laboratoire de neurophysiologie dans lequel il dressera des cartes de connexions entre les diverses zones du cerveau. Il quitte Yale en 1941 pour l'université de l'Illinois à Chicago où il prend contact avec le comité, créé par Nicolas Raskevsky, groupe à la pointe de la recherche en biophysique. Raskevsky défend le développement des modèles mathématiques dans les procédés biologiques.

Walter Pitts, beaucoup plus jeune que McCulloch, à la fin des années 30 assiste en candidat libre aux cours de logique et de biophysique dispensés par Raskevsky à l'université de Chicago. Il intègre peu de temps après le comité et rencontre McCulloch. Mcculloch et Pitts collaborent deux ans pour mettre en place leur théorie logique de l'activité du système nerveux et publient « A logical calculus of the ideas immanent in nervous activity ». De par la proximité et l'intensité de leur travail, ils deviennent des amis intimes jusqu'à leur mort en 1969. Dans leurs hypothèses, les réactions d'une partie du système nerveux afférent la réaction d'un neurone. L'activité impulsionnelle globale du réseau neuronal permettrait d'inférer et d'expliquer l'état mental d'une personne (sensation, idées et relations épistémiques). McCulloch ne met pas de côté sa théorie du Psychon. Ils considèrent que les relations entre les neurones peuvent être modélisées par des fonctions logiques. McCulloch souhaite modéliser l'esprit par une activité neuronale « tout ou rien » du cerveau. De plus les compétences de Pitts en logique permettent de développer et d'écrire cette théorie exposée dans l'article de 1943. McCulloch s'est inspiré des travaux d'Alan Turing. Dans une discussion publique, il indique que la lecture de l'article de Turing sur sa machine l'a orienté dans la bonne direction. Toutefois, une idée fausse s'est propagée dans la communauté scientifique selon laquelle McCulloch et Pitts auraient démontré l'équivalence entre leur réseau et la machine de Turing. Malgré leurs convictions, ils n'ont pas fourni de résultat permettant de mettre en évidence la puissance de calcul de leur réseau. La théorie de McCulloch et Pitts se compose de 5 hypothèses :

- L'activité du neurone est un processus « tout ou rien ».
- Un certain nombre de synapses doivent être excitées pendant la période latente d'adition afin d'exciter le neurone. Ce nombre est indépendant de l'activité des neurones antérieurs et de leur position.
- Le seul délai du système de neurone est le délai synaptique.
- L'activité d'une synapse d'inhibition empêche l'excitation du neurone à ce temps.

- La structure du réseau ne change pas dans le temps.

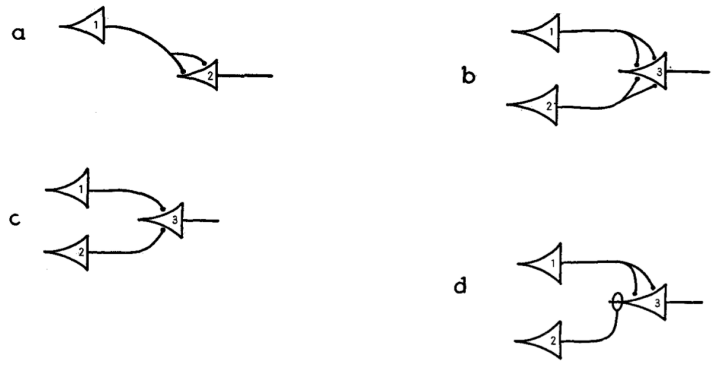


FIGURE 2.9 – *Exemple de circuits électriques modélisant un petit réseau de neurone par Walter Pitts et Warren McCulloch 1943*

On notera une ressemblance entre le neurone formel de McCulloch et Pitts et le neurone biologique. Le processus logique « tout ou rien » correspond au potentiel d'action envoyé sur l'axone. Le nombre de synapses pour activer le neurone se rapproche du seuil de déclenchement du neurone. Par contre, une seule synapse inhibitrice gèle le neurone. Le temps joue un rôle important dans la propagation de l'information à travers le réseau. La dernière hypothèse indique l'impossibilité du réseau d'apprendre. L'état du réseau au temps t dépend de la configuration du réseau au temps t-1. Aucune modification n'est autorisée et donc aucune adaptation.

La théorie de McCulloch et Pitts est le point de départ à la fois du connexionnisme et de l'approche actuelle de l'intelligence artificielle. Elle a fortement inspiré le travail de Norbert Wiener et John Van Neumann. Toutefois, la modélisation de McCulloch et Pitts de leur neurone formel est simpliste. La vision statique des connexions du réseau est en dehors de toute réalité. Des connexions se créent et se défont continuellement dans le cerveau n'engendrant pas obligatoirement des désordres psychologiques. L'approche connexionniste, consistant à considérer que la cognition est le fruit de la connexion d'unité élémentaire qui interagissent entre elles (la pensée humaine ne possède pas à une suite de déduction de logiques, mais est constituée d'automates très simples inter-connectés - Réseau de neurones) s'oppose au cognitivisme (la pensée est un processus de traitement séquentiel d'informations symboliques dirigé par une centrale de contrôle - Machine de Von Neuman/ Ordinateur) [30].

À l'époque à laquelle les réseaux de neurones formels sont introduits par McCulloch et Pitts, les psychologues commencent à tenter de dégager la structure sous-jacente de l'apprentissage et de la mémoire. En 1943, Clark Lenonard Hull (1884 – 1952), psychologue américain, soumet l'idée selon laquelle le mécanisme de la mémoire est double. Il implique un stockage à court terme (Mémoire à court terme) et un à long terme (Mémoire à long terme). À la fin des années quarante, Donald Hebb *(1904-1985)* complète les travaux de Hull et tente de combler le fossé séparant la psychologie et la neurophysiologie. Pour lui, les boucles fermées et les effets de réverbération du signal des neurones expliqué par

McCulloch et Pitts sont susceptibles d'intervenir uniquement dans les mécanismes de la mémorisation à court terme. L'instabilité de ces réverbérations ne peut pas expliquer le stockage des informations à long terme. Il fut le premier à indiquer que le réseau neuronal doit pouvoir se modifier et se renforcer pour la mémoire à long terme. Dans son ouvrage *The organization of behavior* [31], il citera la célèbre règle de Hebb relative au renforcement ou l'atténuation des connexions neuronales du cerveau humain lors de l'apprentissage. Il définit ainsi la première règle d'apprentissage. Lorsque deux neurones liés entre eux sont excités en même temps, le lien les reliant se renforce.

"Faisons l'hypothèse qu'une activité persistante et répétée d'une activité avec réverbération (ou trace) tend à induire un changement cellulaire persistant qui augmente sa stabilité. Quand un axone d'une cellule A est assez proche pour exciter une cellule B de manière répétée et persistante, une croissance ou des changements métaboliques prennent place dans l'une ou les deux cellules ce qui entraîne une augmentation de l'efficacité de A comme cellule stimulant B." Donald Hebb 1949

Pour imager simplement cette règle, prenons la fameuse expérience de Pavlov. Un chien est placé dans une enceinte ou les afférences sensorielles sont réduites : isolé du bruit, sans nourriture, sans odeur. Le matin, un plat de viande très appétissante lui est servi. La vision et l'odeur du plat font saliver le chien. L'expérience se répète dans le temps. Le professeur Ivan Pavlov *(1849-1936)* ajoute un stimulus acoustique comme le son d'une cloche avant de nourrir le chien. Après répétition de cette expérience, le chien salive en entendant le stimulus acoustique sans sentir ou voir son repas. Pavlov parle de réflexe conditionné. Cette expérience met en évidence le renforcement de certaines connexions neuronales décrites par Donald Hebb.

Perceptron

À l'aube des réseaux de neurones, une attention particulière est prise pour incorporer les règles de Hebb dans les structures artificielles proposées. Ces modèles neuronaux s'appuient sur le modèle tout ou rien ou binaire de McCulloch et Pitts. Toutefois, il émerge des modèles adaptatifs, c'est dire dont les poids ou les connexions varient, permettant un apprentissage. Franck Rosenblatt, Psychologue américain, dans son livre *Principle of neurodynamique : Perceptrons and the theory of the brain mechanisms* [32] propose un modèle général nommé "le perceptron". Le perceptron correspond à une brique de base des réseaux de neurones artificiels. L'innovation essentielle réside dans l'introduction de poids numériques adaptables entre les entrées et la sortie du perceptron. L'algorithme d'apprentissage permettant l'ajustement des paramètres du perceptron (poids et biais) est développé par Rosenblatt dans les années soixante. Sa procédure s'approche fortement de la rétro-propagation de l'erreur dans les réseaux utilisés de nos jours pour calculer les paramètres des réseaux. Technique popularisé et dépoussiéré par Rumelhart, Hinton et Williams dans l'article technique *Learning internal representation by error propagation* [33], mais également utilisé par Werbos dans l'article *New Tools for Prediction and Analysis in the Behavioral Sciences* [34] , Yan Lecun dans le document *une procédure d'apprentissage pour réseaux à seuil asymétriques* [35] et l'article de Parker *learning logic* [36]. Le perceptron de Rosenblatt a été analysé

et améliorée par Minsky et Papert [Minsky, M., and S. Papert (1969), *Perceptrons : An Introduction to Computational Geometry* [37]. Le perceptron est un classifieur, il sépare linéairement l'espace en deux parties. Dans un espace à deux dimensions, le séparateur est une droite, un plan dans un espace à trois dimensions et un hyperplan dans un espace supérieur à trois dimensions. Les poids des connexions et le biais permettent de modifier les caractéristiques du séparateur. La sortie du perceptron peut prendre uniquement deux valeurs par exemple '0' ou '1'. Comme le fait la soma dans le neurone biologique, il réalise une simple sommation des i entrées X_i coefficientées par les poids des liaisons et ajoute le biais. À la suite de cette addition, une fonction non linéaire est appliquée pour obtenir la sortie y. Cette fonction, nommée fonction d'activation f correspond à l'action que réalise le neurone biologique lorsque la soma décide d'après les potentiels excitateur et inhibiteur d'activer ou non le neurone et d'envoyer sur l'axone le potentiel d'action. La règle d'apprentissage des poids W_i est très proche de la règle de Hebb. C'est un apprentissage supervisé qui contrairement à la règle de Hebb prend en compte l'erreur observée entre la sortie du perceptron et celle attendue.

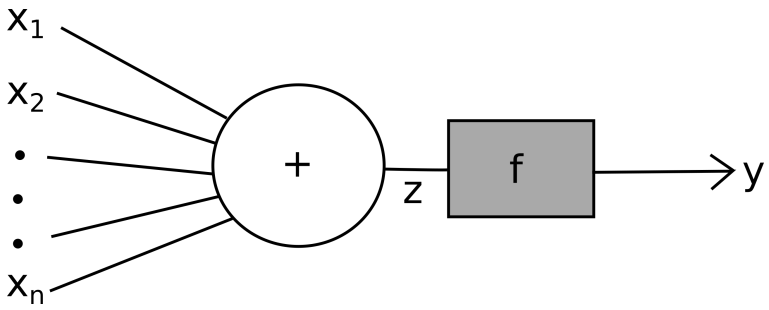


FIGURE 2.10 – *Modèle du perceptron*

L'équation mathématique du modèle du perceptron est donnée ci-dessous :

$$z = \sum_{i=1}^{n} x_i W_i + b \quad et \quad y = f(z) = f\left(\sum_{i=1}^{n} x_i W_i + b\right)$$

Les poids W_i sont des réels positifs ou négatifs, mimant les neurotransmetteurs excitateurs ou inhibiteurs.

Algorithme d'apprentissage des poids pour le perceptron :

1. Initialiser des poids avec des petites valeurs aléatoires
2. Présenter un vecteur x en entrée et calculer la sortie y du perceptron
3. Ajuster les poids des neurones du perceptron en utilisant l'équation ci-dessous

$$w_i'(t+1) = w_i + \eta(d-y)x_i$$

- w'_i poids corrigé
- η taux d'apprentissage
- d sortie désirée
- y sortie calculée
- x_i ième entrée du perceptron reliée au poids w_i

4. Reprendre les opérations de 2 et 3 le nombre d'itérations choisies

La fonction d'activation f joue un rôle extrêmement important. C'est elle qui va décider d'activer ou non la sortie y du neurone. Cette fonction peut être continue ou discontinue. Sans fonction d'activation, la sortie est une simple somme pondérée des signaux entrant dans le neurone. Cette transformation linéaire est simple à résoudre, mais ne pourra pas résoudre des problèmes complexes. Un réseau de neurones sans fonction d'activation se limite à un problème de régression linéaire. La fonction d'activation permet d'ajouter de la non-linéarité et ainsi résoudre des problèmes plus complexes. Afin d'entrainer correctement les réseaux de neurones, la fonction d'activation doit impérativement être dérivable pour réaliser la rétro-propagation du gradient. La sortie y n'est pas la plupart du temps un 1 ou un 0 comme le ferait un neurone biologique. Le neurone biologique envoie des impulsions via un potentiel d'action vers les neurones suivants. Le neurone biologique est actif ou inactif et agit comme un dispositif binaire. Toutefois, en moyennant dans le temps les impulsions on peut modéliser la sortie avec un nombre réel correspondant au nombre d'impulsion dans le temps. Plus il y a d'impulsions et plus le neurone est excité. Les fonctions d'activation réalisent ce travail de quantification de l'excitation du neurone.

Le réseau du perceptron va inspirer les chercheurs dans la recherche de réseau de plus en plus importants. La grande quête de l'apprentissage profond sera ralentie par des obstacles dus au manque de puissance de calcul et de mémoire des ordinateurs. Elle reprendra à la fin de 20eme siècle avec l'évolution exponentielle de la performance des ordinateurs et des visionnaires comme Yan Lecun, Geoffrey Hinton ou Yoshua Bengio.

2.2 Approche Classique de la détection du feu et de la fumée

2.2.1 Le feu et la fumée

Le feu est issu d'une réaction chimique d'oxydoréduction fortement exothermique. La combustion d'un support ne peut avoir lieu uniquement en présence de 3 éléments indiqués dans le le triangle du feu *figure 2.11* :

- Comburant : la plupart du temps il s'agit de l'air et plus particulièrement de l'oxygène présent dans sa composition.
- Combustible : Le support comme du bois, un gaz, des hydrocarbures, ou toute autre matière carbonée comme les matières plastiques...
- Chaleur

FIGURE 2.11 – *Triangle du feu*

La suppression d'un de ces éléments arrête le phénomène de combustion. La fumée est issue d'une combustion incomplète qui a lieu lorsque la quantité de comburant est insuffisante pour permettre une réaction complète du combustible. Dans ce cas des résidus sous forme de matière carbonée (suie, goudron, cendres) sont envoyés dans l'atmosphère générant la fumée. À cette fumée s'additionnent des gaz toxiques pour l'homme comme de l'oxyde d'azote, le dioxyde de carbone, le monoxyde de carbone...

Quant à la flamme, sa couleur dépend principalement du type de combustible et de sa température de combustion. Une combustion incomplète va engendrer une couleur de flamme rouge indiquant une faible température. Les transitions colorimétriques vers l' orange, jaune et blanc mettent en évidence une augmentation de la température de la flamme.

La fumée peut prendre une teinte qui se révèle être un indicateur important du type de combustible présent. Une teinte plutôt blanche indique la présence d'humidité du combustible. La couleur change dès que le matériau s'assèche. De la végétation commence par créer une fumée blanche, dès que l'humidité a été évaporée, la couleur de la fumée tourne vers le brun, marron *Figure 2.12*. Toutefois, la couleur de la fumée dépend également de l'illumination de la scène, des conditions atmosphériques et de la qualité de l'image acquise. Tous ces paramètres rendent la détection de la fumée plus ardue que celle du feu.

FIGURE 2.12 – *Exemple de couleurs de flammes sur un feu de forêt. © Getty / Daryl Pederson*

2.2.2 Détection du feu par colorimétrie

L'illumination d'une scène comportant de la fumée et/ou du feu varie en fonction de la distance entre le feu et l'observateur. De plus, la multiplicité des capteurs et des algorithmes de contrôle des couleurs et illumination embarquée par les caméras implique des différences colorimétriques non négligeables entre plusieurs caméras pour une même scène.

Malgré ces aléas liés à la captation des couleurs de chaque pixel d'une image, la détection du feu et de la fumée par l'analyse colorimétrique reste la première méthode utilisée. La raison principale provient simplement du fait que les caméras codent les images dans un système de couleur RGB (Red Green Blue). Ce codage reprend par mimétisme le codage des couleurs dans l'oeil humain avec ses trois types de cônes de la rétine respectivement rouge, vert et bleu.

La première idée consiste à rechercher des zones de l'image possédant une teinte à dominante rougeâtre. Une règle colorimétrique élémentaire mettant en évidence une surreprésentation du canal rouge par rapport aux canaux vert et bleu comme (R>G>B) permet de détecter du feu dans des environnements sans perturbation notables *Figure 2.13*.

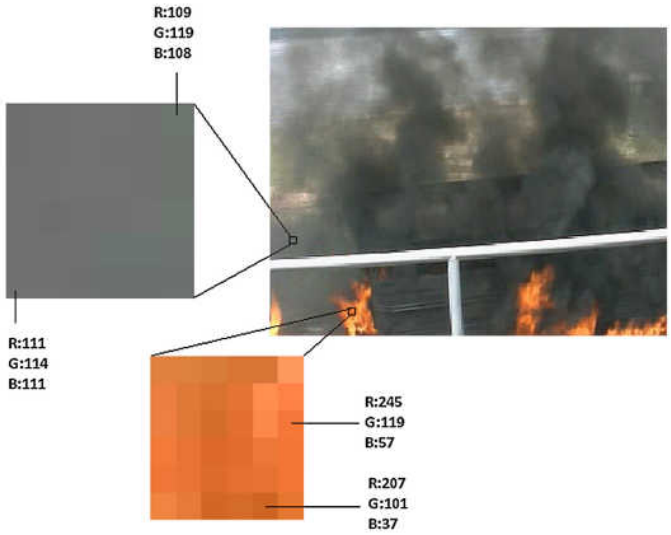


FIGURE 2.13 – *Détection du feu et de la fumée par la couleur des pixels [1]*

Thou-Ho et al [38] ont utilisé des règles chromatiques plus complexes constituées de seuils pour discriminer les pixels du feu des autres pixels d'une image (voir ci-dessous).

- Règle 1 : $R < R_T$ avec R_T un seul colorimétrique fixé par l'expérience
- Règle 2 : R>=G>B
- Règle 3 : $S \geqslant \frac{(255-R)\times S_T}{R_T}$ avec S la valeur de saturation du du pixel et S_T un seuil de saturation fixé.
- **Si** (règle 1) et (règle 2) et (règle 3) sont vraies **Alors** le pixel est considéré comme un pixel de feu, **Sinon** pixel non feu.

Les seuils définis sont fixés de manière empirique à l'aide d'une expertise indispensable dans le feu et la fumée.

Initialement, la fumée sur une image a été détectée en utilisant des méthodes se basant sur ses propriétés colorimétriques. Elle peut être repérée sur une image par des pixels proches du gris avec des valeurs similaires sur les trois canaux R,G,B *Fig 2.13*. La faiblesse de cette technique réside dans le fait que la combustion de divers matériaux va générer des fumées de couleurs variées tirant vers l'orange, le bleu et même le vert provocant une inégalité des trois canaux fondamentaux de couleurs.

Cette difficulté de détection du feu et de la fumée a amené certains chercheurs à explorer d'autre système de représentation colorimétrique comme le HSV, Lab,YUV... Le système HSV (Hue Saturation Value) définit une couleur par sa teinte représentant la nuance de la couleur, la saturation donnant une indication sur son intensité et la valeur qui estime sa brillance. La fumée, même si elle possède une légère couleur, possède comme particularité d'avoir une faible saturation.

L'observation de la décroissance de la chrominance dans les zones contenant du feu a poussé les chercheurs à utiliser le système de couleur YUV. Dans ce système colorimétrique, Y code la luma et U et V représentent la chrominance [4]. Un exemple de conditions dans

l'espace des couleurs YUV pour détecter un pixel appartenant à de la fumée est le suivante :

- Condition 1 : $Y > T_Y$
- Condition 2 : $|U - 128| < T_U$ and $|V - 128| < T_V$

Les seuils T_U, T_V et T_Y sont fixés par l'expérimentation [3].

Les techniques colorimétriques de détection du feu et de la fumée sont imprécises et produisent de nombreux faux positifs et négatifs.

2.2.3 Détection du mouvement des flammes et de la fumée

De multiples techniques permettent d'extraire d'une vidéo les zones dans lesquelles un objet se déplace ou se transforme. L'utilisation de ces techniques est très utile pour détecter les flammes et la fumée, car ce sont des objets évoluant dans le temps, assimilables à des objets en mouvement du point de vue de la modification des pixels. Ces méthodes imposent une caméra fixe.

Soustraction d'arrière-plan (Background subtractor)

La soustraction d'arrière-plan (background subtractor) est très utilisée pour définir des objets en mouvement. L'approche la plus simple consiste à déterminer l'arrière-plan de l'image à un temps t B(i,j,t)=I(i,j,t) avec I(i,j,t) l'intensité du pixel (i,j) au temps t. L'opération consiste ensuite à soustraire la nouvelle image à B(i,j,t) pour chaque pixel. Si cette soustraction est supérieure à un seuil T_h fixé à l'avance, on en déduit que le pixel a subi une modification potentielle impliquant son déplacement $|I(i,j,t) - I(i,j,t-1)| > T_h$ [39]. Il est possible afin d'adoucir la détection, de prendre en compte une moyenne de plusieurs images antérieures pour définir l'arrière plan $B(i,j,t) = \frac{1}{n}\sum_{k=0}^{n-1} I(i,j,t-k)$ [40] ou bien la médiane des n images antérieures $B(i,j,t) = mediane\{I(i,j,t-k)\}$ avec $k \in \{0,...,n-1\}$. Ces méthodes sont simples à mettre en oeuvre, rapides à implémenter et souvent peu coûteuses en temps de calcul. Toutefois, la détection est sensible à la vitesse de l'objet à détecter et dépendante d'un seuil fixé pour l'ensemble des pixels de l'image. Seuil qui reste indépendant du temps. La modélisation de l'arrière-plan étant unimodale (seuil unique de détection), ces méthodes ont du mal à gérer correctement des fonds dynamiques.

Les méthodes précitées sont unimodales et sont incapables de réaliser leur tâche lorsque la vidéo subit un changement brutal de luminosité ou un changement dans l'arrière-plan. Un modèle multimodal s'appuyant sur un ensemble de gaussiennes améliorera grandement la détection des objets dans une vidéo. C Stauffer et son équipe [41] développent un algorithme de soustraction de l'arrière-plan basé sur les modèles de mélange de gaussiennes (GMM). Il définit une distribution multimodale sous forme de gaussienne permettant de modéliser l'arrière-plan ainsi qu'un algorithme dynamique de mise à jour de ce modèle. Chaque pixel de l'arrière-plan est modélisé par un mélange de gaussien permettant de prendre en compte les variations temporelles de l'intensité de chaque pixel de l'arrière-plan. La probabilité qu'un pixel x d'intensité I(x) appartienne à l'arrière plan est donnée

par l'équation suivante :

$$P_{Bg}(I(x,t),\mu(x,t),\Sigma(x,t)) = \sum_{i=1}^{K} \frac{w_i(x,t)}{(2\pi)^{\frac{D}{2}}|\Sigma_i(x,t)|^{\frac{1}{2}}} e^{-\frac{1}{2}(I(x,t)-\mu_i(x,t))^T\Sigma_i^{-1}(x,t)(I(x,t)-\mu_i(x,t))}$$

Ou K représente le nombre de gaussiennes et w_i le poids associé à chaque gaussienne ($\sum_{i=1}^{K} w_i(x,t) = 1$). μ_i et Σ_i sont respectivement la valeur moyenne et la matrice de covariance de la i^{eme} gaussienne pour le pixel courant. Les K modes sont triés de manière croissante suivant le rapport $\frac{w_k}{\sigma_k}$. Un pixel de l'image est assimilé à un pixel d'arrière-plan si sa valeur correspond à l'une des B premières distributions calculées par la relation suivante :

$$B = \underset{b}{argmin} \sum_{i=1}^{b} w_i(x,t) > T$$

avec i l'indice de la gaussienne. Ceci implique que les gaussiennes de variance étroite et de poids important seront celles qui seront majoritairement attribuées à l'arrière-plan. Les nouveaux éléments introduits comme un objet arrivant dans la scène auront quant à eux un faible poids et une forte variance .

Le principal avantage de cette méthode réside dans le fait que chaque pixel de l'arrière-plan possède une historique. Des améliorations ont été apportées à cette méthode comme un modèle GMM adaptatif [42], un contrôle dynamique du nombre de gaussienne [43] ou l'ajout d'un mécanisme bayésien de maximisation de vraisemblance [44].

Toutes ces méthodes de soustraction d'arrière-plan ont été mises en oeuvre dans la détection du feu et de la fumée dans des vidéos [45],[46], [4], [47], [48], [49], [50], [51], [52], [53]

Flot optique (Optical flow)

Deux images successives d'une vidéo donnent des informations sur la dynamique des objets qui s'y déplacent. Le feu et la fumée sont également des objets dont l'analyse dynamique permet de les détecter et de donner une information sur leur vélocité. Le flot optique ou optical flow en anglais décrit le taux directionnel et temporel de pixels dans deux images successives d'une vidéo. Il est possible de définir pour chaque pixel I(x,y,t) de l'image une vitesse. Dans le cas d'une luminosité constante autour d'un pixel déplacé entre deux images successives on estime que sont intensité se conserve $I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t)$. En appliquant les séries de Taylor on obtient une loi linéarisé de la forme : $\nabla I \times \overrightarrow{v} + \frac{\partial I}{\partial t} = 0$ avec ∇I est le gradient spatial de l'intensité de la luminosité, $\overrightarrow{v}$ le flot optique correspondant à la vitesse du pixel. Ces méthodes sont très sensibles à la variation aux bruits dans l'image et demandent une caméra fixe. Elles sont la plupart du temps non compatibles avec le temps réel à cause des calculs complexes exigés. Il existe plusieurs méthodes dont celles de Horn-Schunck [54] et Lucas-Kanade[55] permettant de résoudre cette équation, toutes basées sur la minimisation d'une fonction.

Le flot optique associé à d'autres techniques a été utilisé dans la détection du feu et de la fumée [56], [57], [2] *Figure 2.14*

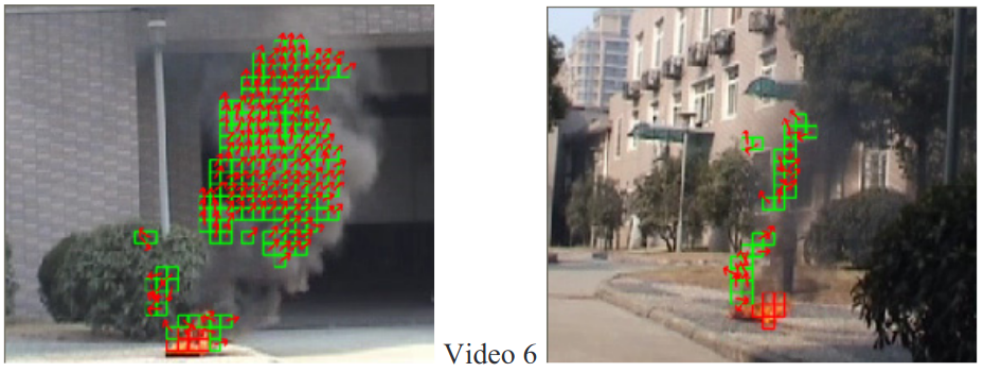


FIGURE 2.14 – *Exemple de détection du feu et de la fumée par flow optique. [2]*

2.2.4 Scintillement et ondelettes

La prise en compte de l'aspect temporel a grandement amélioré la précision de la détection du feu et de la fumée sur des vidéos, car ces objets sont "vivants". Leur forme et les aspects colorimétriques fluctuent dans le temps. La littérature est constituée de nombreuses méthodes de détection du feu et de la fumée dans des vidéos [38], [58], [59]. Ces méthodes utilisent entre autres la signature colorimétrique propre du feu, mais également son déplacement et sa géométrie. S Chen et al. [38] utilisent, en plus des paramètres colorimétriques une méthode permettant de détecter le scintillement de la flamme. L'observation à la limite d'une région de flammes a montré une augmentation de la fréquence de scintillement entre 0,5Hz et 20Hz (ces fréquences fluctuent en fonction de la fréquence d'échantillonnage de la vidéo) *Figure 2.15*. Ce scintillement caractéristique à poussé la communauté scientifique à étudier les images de feu dans le domaine de fréquentiel pour détecter les contours [60]. Toutefois, un feu démarrant, la plupart du temps, produit des instabilités non linéaires impliquant un comportement chaotique. Le scintillement des pixels aux abords du contour de feu se matérialise sur une large bande fréquentielle. Il devient alors impossible de discerner le contour de la flamme des autres objets de la scène [61]. De plus, la transformé de Fourier comme la Fast Fourier Transform (FFT) perd les informations relatives au temps. Afin de conserver une information temporelle l'utilisation de Short-Time Fourier Transform (STFT) est requise. Cette transformation consiste à subdiviser le signal temporel en fenêtre de taille fixe pour y appliquer une transformée de Fourier. Cette fenêtre temporelle devient un paramètre important pour la détection du feu difficilement paramétrable. L'agrandissement de cette fenêtre améliore la précision fréquentielle en diminuant la précision temporelle des phénomènes. A contrario, la réduction de la fenêtre améliore la précision temporelle en diminuant celle fréquentielle.

Pour pallier à ce problème, Jean Morlet en collaboration avec Alex Grossman ont développé la transformée en ondelette. La détection des contours de la fumée par l'exploitation de l'énergie dans l'image par des ondelettes a été mise en oeuvre par U Toreyn et son équipe [4]. La méthode se présente en cinq étapes :

1. Recherche des zones de modification des pixels en utilisant une soustraction d'arrière-plan [62]. La caméra doit être impérativement fixe.

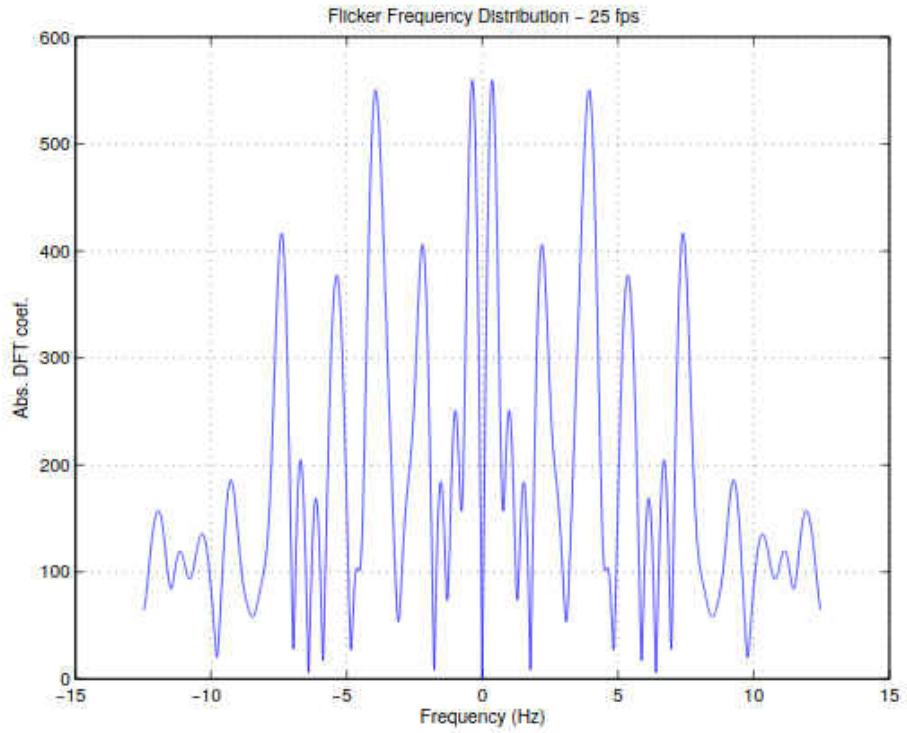


FIGURE 2.15 – *Distribution fréquentielle du scintillement d'un pixel à la limite d'une région de flamme. Fréquence d'échantillonnage de la vidéo : 25 images par seconde. [3]*

2. Utilisation des transformées discrètes en odelettes dans l'espace (DWT) afin de rechercher des régions de décroissances des hautes fréquences. Régions qui peuvent être assimilées à des contours de la fumée. Généralement, les arrêtes et les textures contribue à générer des informations à hautes fréquences de l'image. L'énergie des images secondaires obtenues par les odelettes chute en présence de fumée .
   DWT produit 3 images secondaires à l'aide d'un banc de filtre *Fig 2.17*, appelé horizontale H_t, vertical V_t et diagonal D_t. L'énergie est calculée à partir des images secondaires en divisant l'image I_t en blocs b_k.

$$E(I_t, b_k) = \sum_{i,j \in b_k} H_t^2(i,j) + V_t^2(i,j) + D_t^2(i,j)$$

   L'énergie d'un bloc varie significativement dans le temps entre la présence et l'absence de fumée *Fig 2.16*. Cette variation est utilisée pour détecter la fumée.

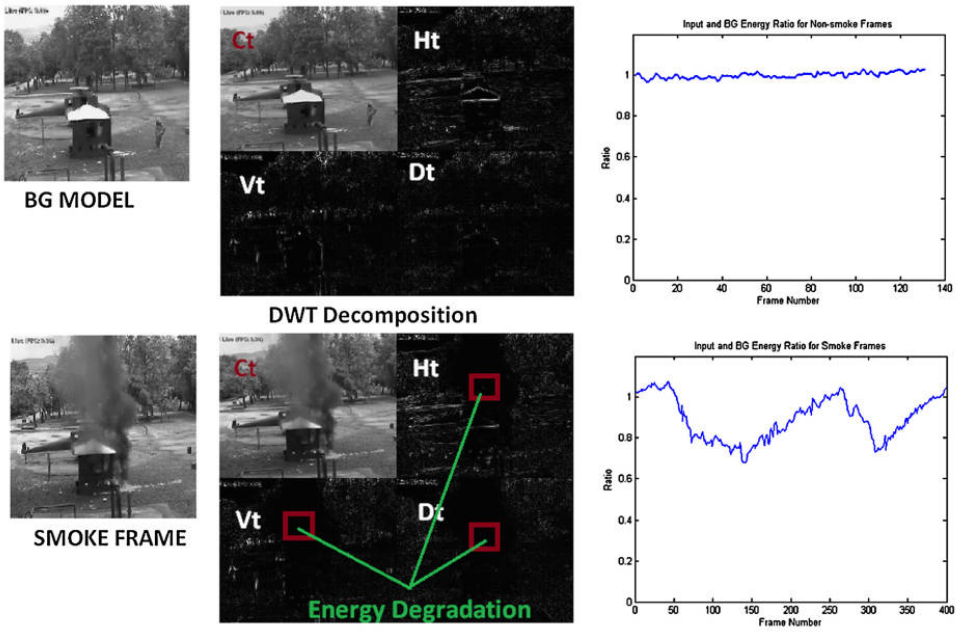


FIGURE 2.16 – *Détection de la fumée par DWT. En présence de fumée, le ratio entre l'énergie due aux odelettes et l'arrière plan diminue. [1]*

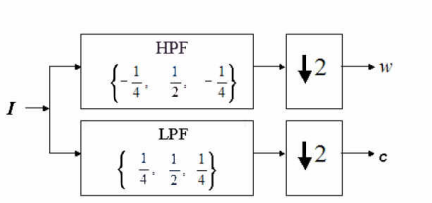


FIGURE 2.17 – *Banque de filtre de la DWT. [4]*

3. Vérification de la décroissance de U et V dans les zones sélectionnées. L'expérience montre que lorsque la fumée s'épaissie, les valeurs de U et V deviennent moins importantes que celles de l'arrière-plan pour les mêmes zones.
4. Analyse du scintillement par un Modèle de Markov Caché (HMM) dont les coefficients ont été tirés d'une phase d'entrainement temporelle avec des pixels appartenant à des images de fumée et d'autres n'appartenant pas à la fumée.
5. En plus de l'analyse temporelle et colorimétrique, le contour des potentielles zones de fumée est analysé en utilisant une transformation par ondelette à échelle unique en utilisant une seule fois le banc de filtre *Fig 2.17*.

.

Enfin l'article [63] complète son travail en utilisant un algorithme adaptatif dans lequel les poids sont mis à jour en utilisant la méthode des moindres carrés moyens (LMS) pendant l'apprentissage.

U Toreyin et son équipe réalisèrent un travail important et très intéressant sur le sujet. L'article [47] propose une méthode de détection de feu et de flamme sur des vidéos basées

sur la dynamique des pixels. Il utilise une méthode hybride de détection d'arrière-plan pour récupérer les zones d'activité dans chaque image. Ces zones de pixel sont comparées à des distributions colorimétriques d'images de références de feu, puis une analyse temporelle est utilisée pour déterminer l'activité de scintillement. Le signal de fréquence f de scintillement des pixels contenant des zones potentiellement de feu est envoyé dans deux filtres : un filtre passe haut et un filtre passe bas. Les signaux de sortie des filtres sont analysés afin de déterminer si les pixels appartiennent à une flamme ou à un objet ordinaire possédant les caractéristiques colorimétriques du feu. Une quatrième étape consiste à procéder à une analyse spatiale par ondelettes des pixels contenant du feu afin de capturer la variation colorimétrique des pixels. Les zones contenant des objets ayant des caractéristiques colorimétriques proches du feu possèdent généralement une faible variation colorimétrique spatiale comparée aux zones contenant des flammes. Une fusion des quatre étapes expliquées ci-dessus va permettre de localiser les zones contant du feu. Cette méthode requière une fréquence d'échantillonnage minimale de la vidéo afin d'acquérir correctement la fréquence de scintillement des pixels du feu et ne peut pas fonctionner sur des images statiques. Ses travaux relatés dans l'article [64] ont eu pour objectif d'améliorer sa méthode en utilisant des modèles de Markov séparé pour les pixels reliés à une flamme et ceux d'autres autres objets.

T Celik et son équipe dans l'article [65] utilisent un algorithme basé sur le système de couleur YCbCr à la place du célèbre système RGB afin de séparer la luminance et la chrominance dans les images pour segmenter les flammes. L'article [48] apporte des améliorations sur la segmentation des flammes dans des images, toujours dans l'espace de couleur YCbCr, en utilisant de logique floue. T Celik est le premier [48] à mettre en oeuvre une détection du feu et de la fumée dans des images.

Ces méthodes décrites ci-dessus ont fait avancer la recherche dans le domaine de la détection et la segmentation du feu et de la fumée dans des images et/ou des vidéos. Toutefois, elles restent imparfaites par les nombreux faux positifs ou faux négatifs engendrés. Il faut ajouter que ces méthodes requièrent une expertise dans le domaine colorimétrique des objets à détecter afin de réaliser les vecteurs caractéristiques ou les règles à observer dans les algorithmes. Afin de se libérer des défauts de ces méthodes, la communauté scientifique s'est lancée, pour la détection du feu et la fumée dans des méthodes basées sur des réseaux de neurones et l'apprentissage profond.

2.3 Réseau convolutif et bio-inspiration

Dans cette section, nous aborderons la détection du feu et de la fumée en utilisant des réseaux de neurones convolutifs (CNN). Ces réseaux convolutifs se sont grandement inspirés du fonctionnement biologique de la fonction visuelle chez les mammifères allant de l'acquisition de l'information visuelle par la rétine jusqu'au traitement de l'information dans le cortex visuel. Je vous propose de définir en premier lieu le champ réceptif visuel dans les CNN et leur forte bio-inspiration pour passer ensuite à l'évolution des différents réseaux convolutifs. Enfin, nous nous attarderons sur les applications liées à la détection du feu et de la fumée avec ce type d'apprentissage.

2.3.1 Champ réceptif visuel et convolution

Le terme "champ réceptif" est tiré de la biologie et plus particulièrement de la médecine. Il définit une zone de l'espace sensoriel connecté à des neurones. La première utilisation de ce terme technique est à mettre au profit du neurophysiologiste Charles Scott Sherrington [66] dans son article de recherche de 1906 relatif "scratch reflex" de la peau d'un chien. Il fallut attendre une trentaine d'années pour que le concept de champ réceptif émerge dans le monde scientifique entre autres par l'enregistrement d'Haldan Keffer Hartline [67] [68] de la réponse d'un neurone à une stimulation sur la rétine d'un vertébré. Hartline définit le champ réceptif ou " receptive field" la zone de la rétine à stimuler pour obtenir une réponse de décharge sur un neurone du nerf optique.

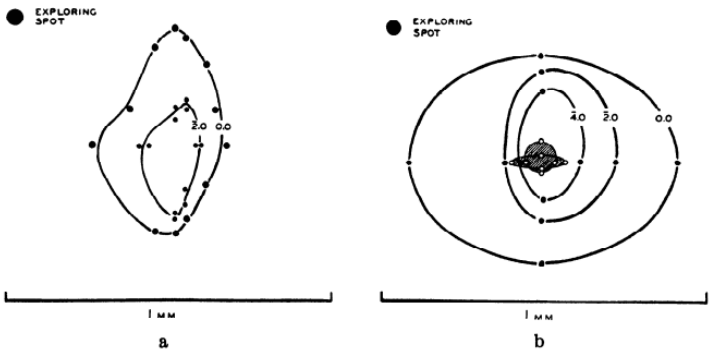


FIGURE 2.18 – *Hartline figure - research article "THE RECEPTIVE FIELDS OF OPTIC NERVE FIBERS" 1940 - Charts of the retinal regions supplying single optic nerve fibers (eye of the frog). a. Determination of the contours of the receptive field of a fiber at two levels of intensity of exploring spot. Dots mark position8 at which exploring spot (50 p diameter) would just elicit discharge8 of impulses, at the intensity whose logarithm is given on the respective curve (unit intensity = 2.104 meter candles). No responses at log I = -3.0, for any location of exploring spot. This fiber responded only at "on" and "off." b. Contour8 (determined by four points on perpendicular diameters) of receptive field of a fiber, at three levels of intensity (value of log I given on respective contours). In this fiber steady illumination (log I = 0.0 and -2.0) produced a maintained discharge of impulses for locations of exploring spot within central shaded area ; elsewhere discharge subsided in l-2 seconds. No maintained discharge in response to intensities less than log I = -2.0 ; no responses at all to an intensity log I = -4.6.*

Il existe une hiérarchisation des champs réceptifs dans le cortex humain. Les informations provenant des cellules sensorielles photosensibles (cônes et bâtonnets) transcrivent une information lumineuse en potentiel d'action. Les cônes récupèrent l'information visuelle relative aux couleurs, tels des pixels qui codent l'information visuelle sur trois canaux rouge, vert et bleu. Les bâtonnets quant à eux très sensibles permettent d'envoyer une information sur la luminosité ambiante. Les potentiels d'action envoyés par les cellules photosensibles sont prè-traités au sein même de la rétine principalement par des neurones bipolaires et ganglionnaires. Le champ réceptif de l'information visuelle varie en fonction de sa position spatiale. Le champ réceptif est très petit dans la fovéa et augmente en dehors de celle-ci.

L'information visuelle sous forme de potentiels d'actions quitte l'œil via le nerf optique. Elle passe par le chiasma optique ou s'opère une redistribution entre les nerfs optiques issus de l'œil droit et gauche. Cette redistribution dans le chiasma permet une séparation des hémichamps droit et gauche. L'hémichamp droit est géré par l'hémisphère cérébral gauche et l'hémichamp gauche par l'hémisphère cérébral droit. Chaque voie passe par le corps genouillé latéral du thalamus constituant un simple relai. L'information quitte le corps genouillé latéral sans avoir subi de modifications substantielles et est envoyée vers le cortex visuel situé dans le lobe occipital du cerveau. Le cortex visuel primaire est composé de multiples aires visuelles hiérarchisées permettant de détecter (voie ventrale) et de localiser des objets (voie dorsale) à partir des informations visuelles perçues.

Le réseau convolutif ou convolutionnel a été fortement inspiré initialement par les travaux de recherches du début des années soixante en neurobiologie de David Hunter Hubel et Torsten Wiesel mettant en évidence les cellules simples répondant plus fortement à des orientations définies et les cellules complexes qui possèdent une plus grande invariance spatiale due au pooling de cellules simples [69]. Travaux repris par Kunihiko Fukushima dans son modèle du neocognitron [70] des années quatre-vingts sont sources d'inspiration à son tour du modèle ConvNet [71] [72] [73]. Le modèle du neocognitron synthétise les connaissances neurobiologiques dans le but de construire un modèle artificiel visuel permettant de reconnaitre des images simples. La structure fait apparaitre les idées de la hiérarchisation des couches *Figure 2.19* ainsi que l'opération de convolution au sein des cartes de caractéristiques d'une même couche *Figure 2.20*.

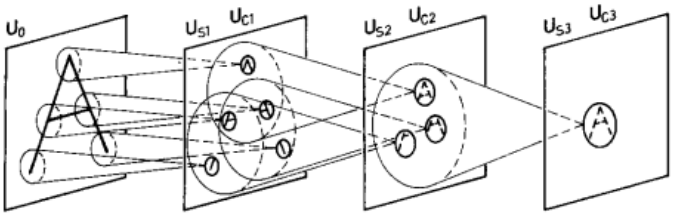


FIGURE 2.19 – *Interconnexion entre les cellules et les réponses dans le modèle du neocognitron de Fukushima*

Les opérations de convolution ainsi que les opérations de pooling sont directement inspirées des cellules simples et des cellules complexes du système visuel des mammifères [71]. Les zones des aires visuelles du cortex (LGN - V1 - V2 - V4 et IT) du système ventral étant assimilées aux diverses couches du réseau de convolution. La hiérarchisation des couches, composées des cartes de caractéristiques, implique une augmentation du niveau d'abstraction de représentation de chaque couche. La première couche représentera les pixels (point de l'image correspondant à l'image d'entrée), la seconde représentera des orientations et donc des segments de droite, la troisième des associations de segments... Ainsi de suite, jusqu'à arriver à l'objet à classifier en sortie de réseau.

Les réseaux convolutifs de par leur organisation et leur constitution possèdent également un champ réceptif visuel. Ils exploitent et conservent dans les couches successives la

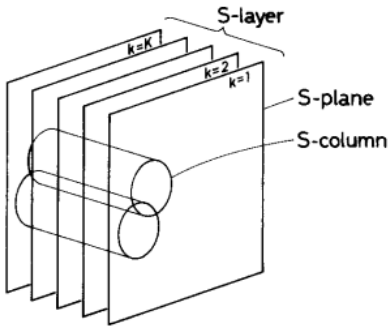


FIGURE 2.20 – *Relations entre les différentes couches dans le modèle du neocognitron de Fukushima*

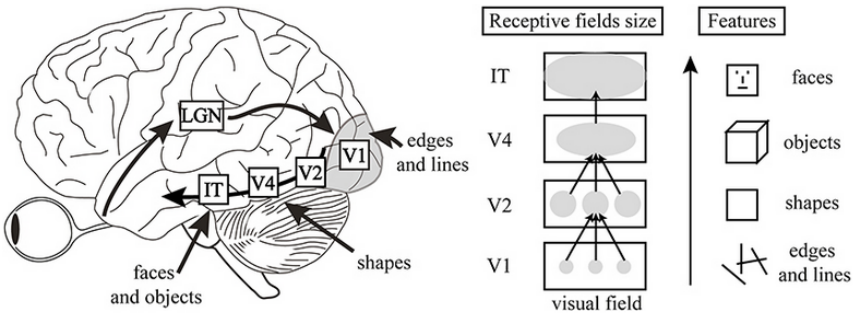


FIGURE 2.21 – *Comparaison du fonctionnement du chemin ventral et du CNN dans les opérations de convolution et de pooling.*

distribution spatiale de l'échantillonnage initial (image d'entrée). L'image originelle codée sur un ou plusieurs canaux subit des opérations de convolutions. Opérations mathématiques simples dans un espace de deux dimensions la plupart du temps, consistant à multiplier un noyau de convolution par les valeurs de l'image à traiter. La taille du noyau de convolution et son pas de déplacement dans les cartes de caractéristiques définissent le champ réceptif visuel de la couche n. Le cas ou l'image possède plusieurs canaux, chaque image possède son noyau de convolution le résultat de l'opération de convolution sera la somme des opérations de convolutions sur chaque canal. La principale différence avec un réseau de neurones classique (perceptron multicouche) provient de la diminution du nombre de paramètres du réseau. Quand un réseau neuronal composé de perceptron multicouche doit avoir tous ses neurones de la couche n-1 connectés avec ceux de la couche n créant un nombre de paramètres d'apprentissage gigantesque pour une image, le réseau convolutionel quant à lui va mettre en commun pour chaque couche le noyau de convolution. Cette particularité réduit fortement le nombre des paramètres d'apprentissage du réseau et assure une vitesse de classification ou de segmentation beaucoup plus rapide. De plus, dans le cas du réseau convolutionel, les cartes de caractéristiques sont obtenues par la translation de noyaux identiques sur l'image ou les cartes de caractéristiques des couches antérieures. Toute l'image ou la carte de caractéristiques est traitée de manière identique dans l'es-

pace, permettant une certaine invariance par translation des caractéristiques des objets à détecter.

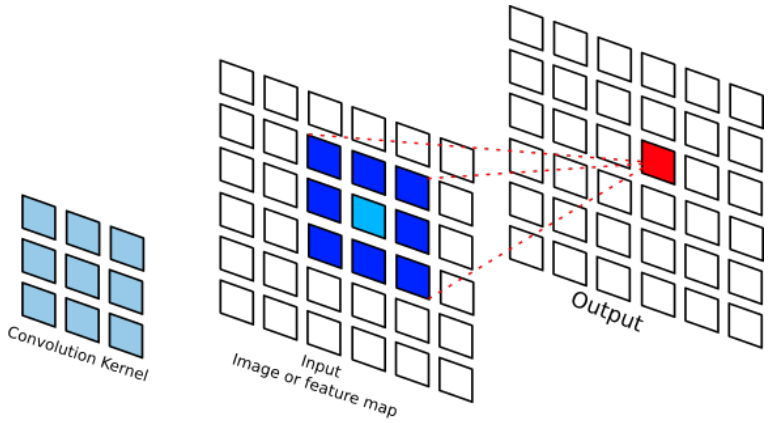


FIGURE 2.22 – *Opération de convolution avec un noyau 3x3 sur une image à un canal.*

La partie bleue foncée sur la *Figure* 2.22 représente la zone prise en compte par l'opération de convolution pour déterminer la valeur de la cellule de la carte de caractéristique de sortie en rouge. Cette zone représente le champ réceptif de l'opération de convolution, c'est-à-dire la zone des valeurs de la carte de caractéristiques d'entrée prise en compte pour calculer une valeur, centrée autour du noyau de convolution, de la carte de caractéristiques de sortie. Plus le noyau de convolution est important et plus la champ réceptif est important. Toutefois, la taille du noyau de convolution influe sur le temps de traitement de l'opération de convolution car toutes les cellules de la carte de caractéristiques d'entrée vont subir l'opération de convolution. Pour un noyau de 3x3, il y aura 9 opérations de multiplication par pixel et par carte de caractéristique d'entrée pour définir les valeurs de la carte de caractéristique de sortie. Pour un noyau 5x5, il y aura 25 opérations mathématiques et pour un noyau 7x7, pas moins de 49 opérations. Au vu du temps de calcul, il faudra réaliser un compromis entre la taille du champ réceptif et le temps de traitement des opérations de convolution.

Le pas de déplacement du noyau de convolution dans la carte de caractéristiques d'entrée ou "stride" est généralement unitaire. Toutefois, il peut prendre des valeurs supérieures à 1 dans des cas spécifiques.

La taille de la carte de caractéristiques de sortie est toujours inférieure à la taille la carte de caractéristiques d'entrée avec $W_o = \frac{W_i - K}{S} + 1$ *(avec W_o la taille de la carte de caractéristique de sortie, W_i la taille de la carte de caractéristiques d'entrée, K la taille du noyau de convolution qui peut être attribué au champ réceptif et S le pas de déplacement du noyau de convolution dans la carte de caractéristiques d'entrée).* Il est possible de contourner ce problème et d'obtenir une identité de taille pour les cartes de caractéristiques d'entrée et de sortie en ajoutant des zéros autour de la carte de caractéristique d'entrée. Operation nommée padding operation ou zero padding. Le padding ne modifie pas le résultat de sortie de l'opération de convolution, toutefois, elle réduit le champ réceptif sur les bords.

Un réseau de convolution est composé d'une succession d'opérations de convolution possédant un champ réceptif dû à la taille du noyau de convolution. Chaque nouvelle

convolution dans le réseau augmente la taille du champ réceptif. La plupart des réseaux convolutifs comme VVG [74] , U-Net, etc. utilisent des successions de convolutions avec des noyaux 3x3. Le résultat de convolution prend en compte l'information se trouvant à 1 pixel dans les 2 directions de l'espace de l'image (largeur et hauteur). Le champ réceptif de chaque cellule de la carte de caractéristiques de sortie est donc de 3 pixels dans la carte de caractéristiques d'entrée. La succession de deux convolutions 3x3 produit une augmentation du champ réceptif comme le montre la figure ci-dessous *figure* 2.23. Dans ce cas, le champ réceptif est de 5x5 cellules par rapport à l'entrée.

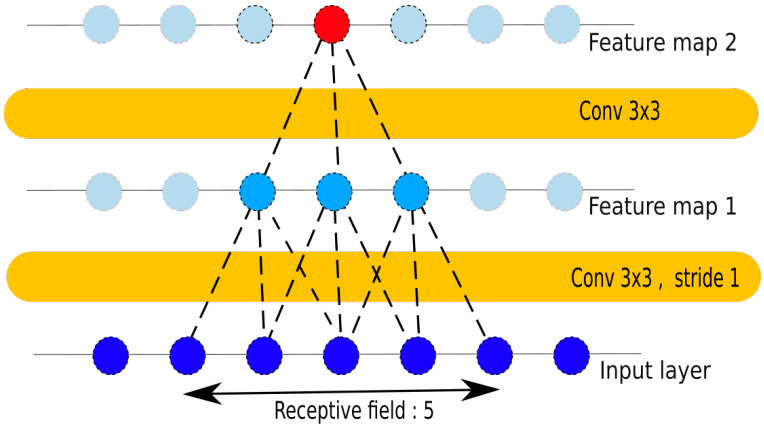


FIGURE 2.23 – *Champ réceptif pour deux convolutions successives 3x3 avec un pas de 1. Le champ réceptif est de 5x5. Les liaisons en pointillées indiquent les cellules prisent en compte sur la couche antérieur pour le calcul de la cellule sur la couche du dessus.*

Le pas de déplacement du noyau de convolution joue un rôle important sur la taille du champ réceptif. La *Figure* 2.24 schématise le champ réceptif avec deux convolutions de noyau 3x3 avec un pas de 2. Le champ réceptif est plus important, mais contrairement au pas unitaire de la *Figure* 2.23, les cellules ne sont pas prises plusieurs fois pour le calcul des valeurs des cellules de la carte de caractéristique 1 *(feature map 1)*

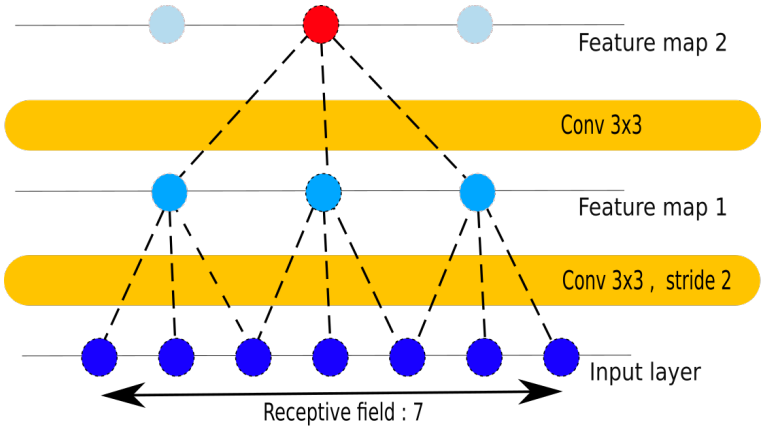


FIGURE 2.24 – *Champ receptif pour deux convolution succeccives 3x3 avec un pas de 2. Le champ receptif est de 7x7*

La majorité des réseaux convolutifs utilise des fonctions permettant de réduire la taille des cartes de caractéristiques. La plus utilisée est le maxpooling qui sélectionne la valeur maximale des données contenues dans une fenêtre. Cette opération permet de réduire le temps de calcul du réseau en diminuant la taille des cartes de caractéristiques. Cette opération permet d'obtenir une certaine invariance à la translation des caractéristiques des objets à classifier ou à segmenter. La *figure* 2.25 propose une mesure du champ réceptif pour une opération de pooling de taille 2x2 et de pas de 2 suivi d'une convolution 3x3 et de pas unitaire.

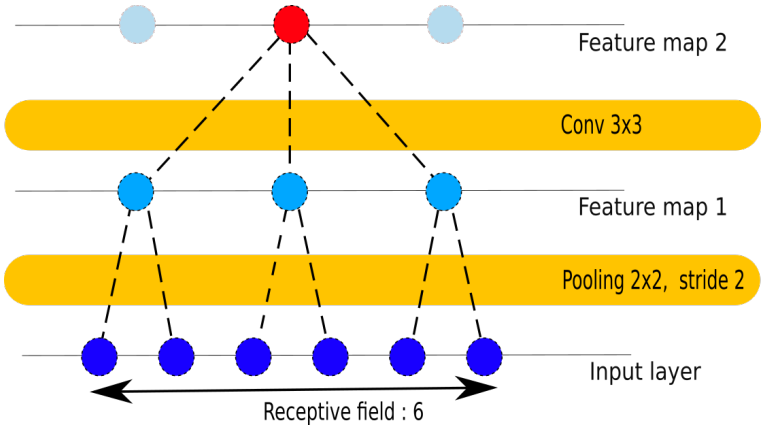


FIGURE 2.25 – *Champ réceptif pour un pooling 2x2 avec un pas de 2 suivi d'une opération de convolution 3x3 avec un pas unitaire. Le champ réceptif est de 6x6*

La plupart du temps les réseaux convolutifs sont constitués dans leur structure de plusieurs opérations de convolutions de faibles noyaux (3x3 par exemple) qui se succèdent permettant de réduire le nombre de paramètres d'apprentissage du réseau. Un noyau de convolution 3x3 demande seulement 9 paramètres d'apprentissage par carte de caractéristiques et possède un champ réceptif de 3 pixels dans la carte de caractéristique. Un noyau

de convolution 5x5 demandera 25 paramètres pour un champ réceptif de 5 pixels. Or, si nous plaçons 2 convolutions de noyaux 3x3 avec un pas unitaire l'une derrière l'autre, le nombre de paramètres par carte de caractéristiques sera de 18 avec un champ réceptif de 5 pixels. On notera le gain de paramètre d'apprentissage et donc de calcul tant pour la phase d'apprentissage que pour la phase d'inférence. Ce gain deviendra primordial lorsque le nombre de couches et/ou le nombre de cartes de caractéristiques sera important.

2.3.2 Évolution des réseaux convolutifs

La détection d'objet, avant la révolution du réseau convolutionnel, consistait à déterminer un ensemble de paramètres regroupés dans un vecteur de caractéristiques propre à cet objet. Les caractéristiques devaient donc être choisies par des experts du domaine à la main. Chaque objet étant différent il demandait des caractéristiques différentes. L'appel à un expert amenait une certaine subjectivité sur le vecteur de caractéristiques choisi. Les objets saillants ou possédant des couleurs caractéristiques fortes permettent plus facilement de déterminer ces vecteurs de caractéristiques. Les objets tels que la fumée qui ne possèdent pas de saillance évidente avec des domaines colorimétriques assez larges sont difficilement caractérisables.

La détermination des objets à détecter pouvait reposer sur des particularités colorimétriques avec des seuils ou des variations de contraste tels que les HOG, SIFT, SURF... Ce vecteur de caractéristiques extrait de l'image était envoyé vers une classifieur composé, par exemple, d'un réseau de neurones afin de déterminer la classe de l'objet présent dans l'image.

Dans le cas d'un réseau de convolution, des images sont présentées au réseau composé de plusieurs couches. Chaque image, dans le cas d'apprentissage supervisé, est reliée à une classe. La dernière couche du réseau se termine par un classifieur qui a pour objectif de déterminer la classe la plus probable de l'objet contenu dans l'image d'entrée. La présence d'un expert définissant les paramètres du vecteur de caractéristiques n'est pas obligatoire. Le réseau lors de la phase d'apprentissage va déterminer tout seul les caractéristiques importantes à prendre en compte pour réaliser une classification performante. Le réseau va trouver des caractéristiques complexes qu'un humain ne peut comprendre et définir. Le premier réseau convolutif nommé LeNet-5 *Fig 2.26* à été mis au point pour détecter des chiffres manuscrits par Yan LECUN [72].

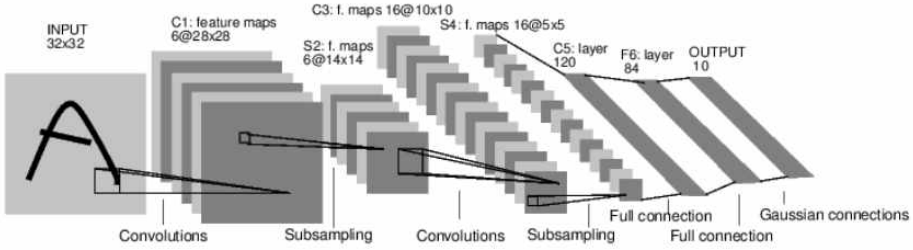


FIGURE 2.26 – *Premier réseau convolutif. Yan LECUN 1990*

Cette découverte va rester dans l'oubli pendant plusieurs dizaines d'années principalement à cause de la faiblesse de la puissance de calcul nécessaire pour entraîner et utiliser ces réseaux. Le CNN renaîtra de ses cendres grâce à l'évolution exponentielle des performances des ordinateurs. Krizhevsky Alex a été le pionnier dans l'architecture profonde avec son réseau AlexNet [75] et [76] *Figure 2.27* donnant une très bonne performance de classification sur la base de données de ImageNet.

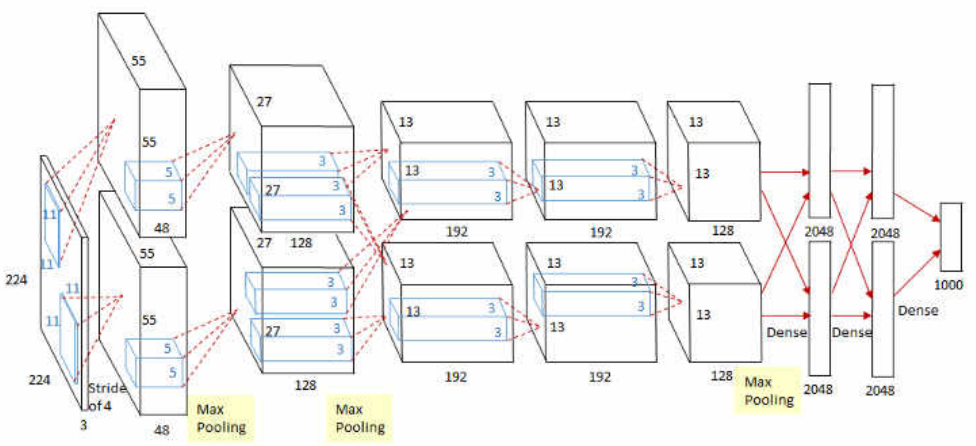


FIGURE 2.27 – *AlexNet réseau*

Le réseau ZFNet[77] à été le vainqueur de la compétition ILSVRC (ImageNet Large Scale Visual Recognition Competition) en 2013 *Fig 2.28*. C'est une reprise du réseau AlexNet avec une optimisation de celui-ci.

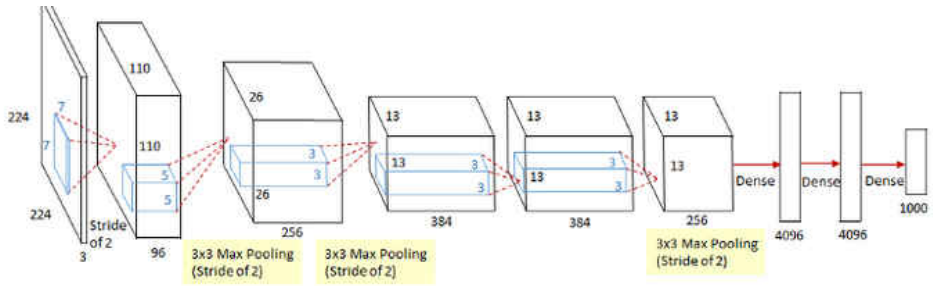


FIGURE 2.28 – *ZFNet réseau*

Le réseau VGG16 [74] Visual geometry group possède 16 couches (13 couches de convolutions et 3 couches totalement connectées) *Fig 2.29*. C'est un modèle qui a été introduit par l'université d'Oxford. Il utilise des convolutions de noyaux 3x3 uniquement.

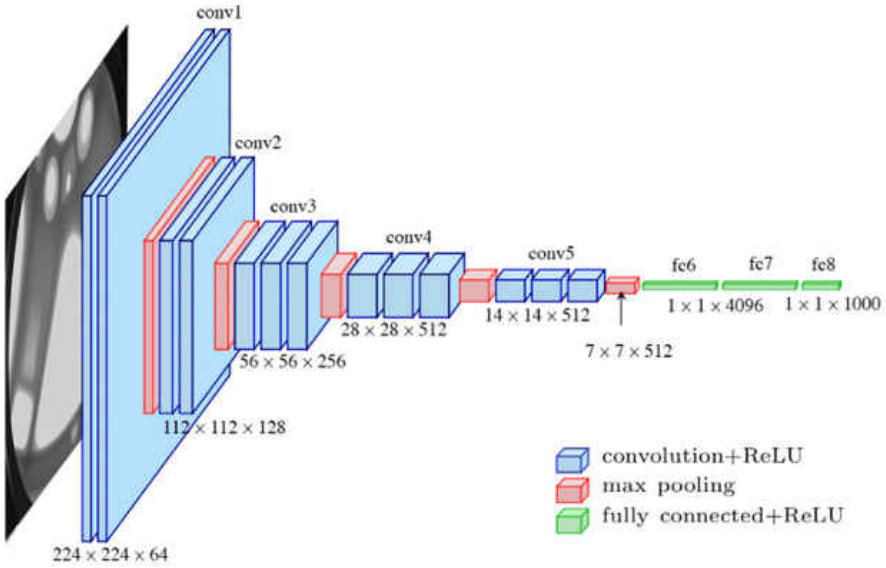


FIGURE 2.29 – *VGG16 réseau*

La particularité du réseau GoogLeNet [78] tient dans la réalisation en parallèle de convolutions avec des noyaux de tailles différentes (1x1, 3x3, 5x5 et 3x3 max pooling) *Fig 2.31* afin extraire différentes caractéristiques entre deux couches. Le résultat des différentes opérations de convolutions est concaténé pour créer la nouvelle carte de caractéristiques. L'utilisation des convolution 1x1 réduit le temps de calcul et permet de se libérer de la taille fixe de l'image d'entrée contrairement aux réseaux AleNet, VGG16 et FZNet. L'optimisation de ce réseau permet d'augmenter le nombre de couches *Fig 2.30*.

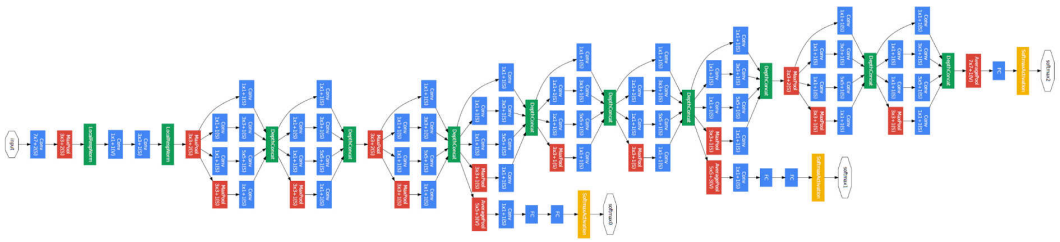


FIGURE 2.30 – *GoogLeNet réseau*

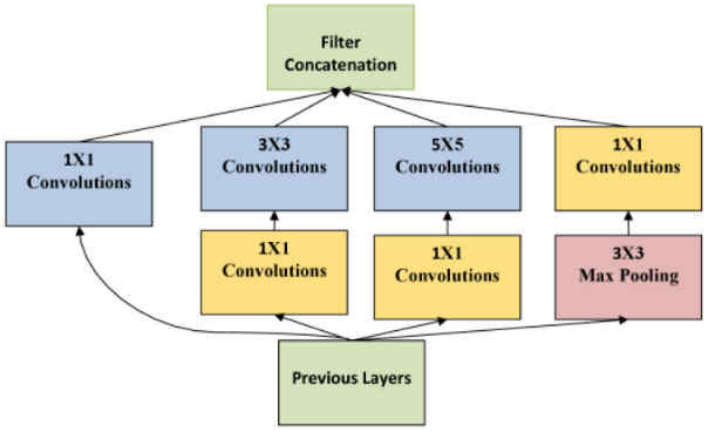


FIGURE 2.31 – *GoogLeNet inception module*

Le module Inception [78] a apporté une plus-value à la classification des réseaux convolutifs. L'idée maîtresse réside dans la réalisation d'opérations de convolutions en parallèle avec des noyaux de tailles différentes *Figure 2.32*. Les cartes de caractéristiques de ces multiples convolutions de tailles de noyaux différentes sont concaténées entre elles. Un pooling est également réalisé avec un pas unitaire avec un noyau 3x3 et vient s'ajouter aux cartes de caractéristiques réalisées avec les opérations de pooling. L'objectif est de pouvoir déterminer les caractéristiques d'un objet de taille différente dans les images.

Afin de réduire fortement le temps de calcul dû aux noyaux de convolution 5x5 et 3x3 une couche supplémentaire dite d'étranglement "bottleneck" est ajoutée *Figure 2.33*. Cette couche est réalisée à partir d'une opération de convolution avec un noyau 1x1. Le nombre de carte de caractéristiques de cette couche est réduit d'où le nom de couche d'étranglement. Cette couche est suivie de l'opération de convolution avec un noyau de taille 5x5 ou 3x3. Cette couche intermédiaire réduit le coût des opérations de calcul d'un facteur 10. La convolution 1x1 qui suit l'opération de pooling a pour seul objectif de réduire le nombre de cartes de caractéristiques de cette voie de 192 à 28. Il existe plusieurs types de versions de ce module d'inception qui ne seront pas discutées ici.

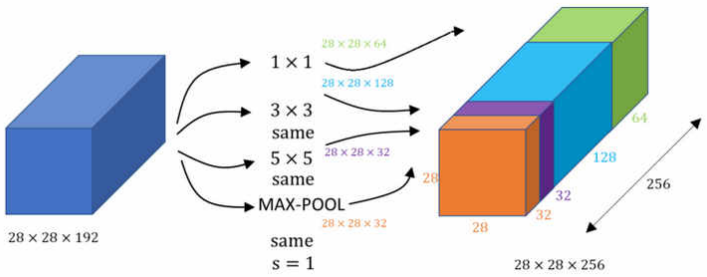


FIGURE 2.32 – *Inception Bloc*

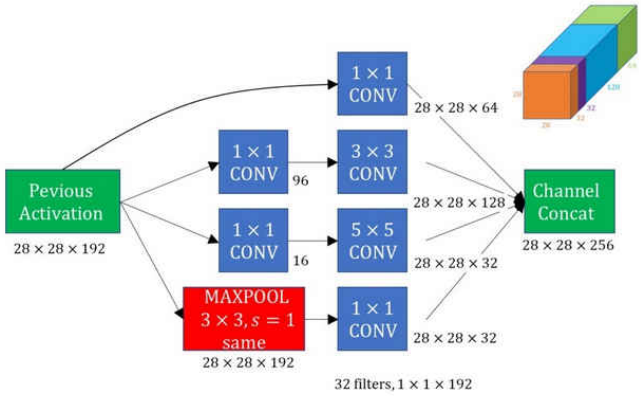


FIGURE 2.33 – *Inception Bloc*

L'ajout toujours croissant de couches (5 couches de convolutions pour AlexNet, 16 pour VGG16 et 22 pour GoogLeNet) complique la tâche de l'entraînement du réseau et peut provoquer la disparition du gradient de rétro propagation stoppant le réglage des poids de celui-ci. L'idée maîtresse de l'architecture ResNet [79] *Figure 2.34* est d'avoir introduit une boucle de connexion identique qui court-circuite les opérations de convolutions *Figure 2.35*. Cette recopie de l'information des couches moins profondes permet d'atténuer la disparition du gradient.

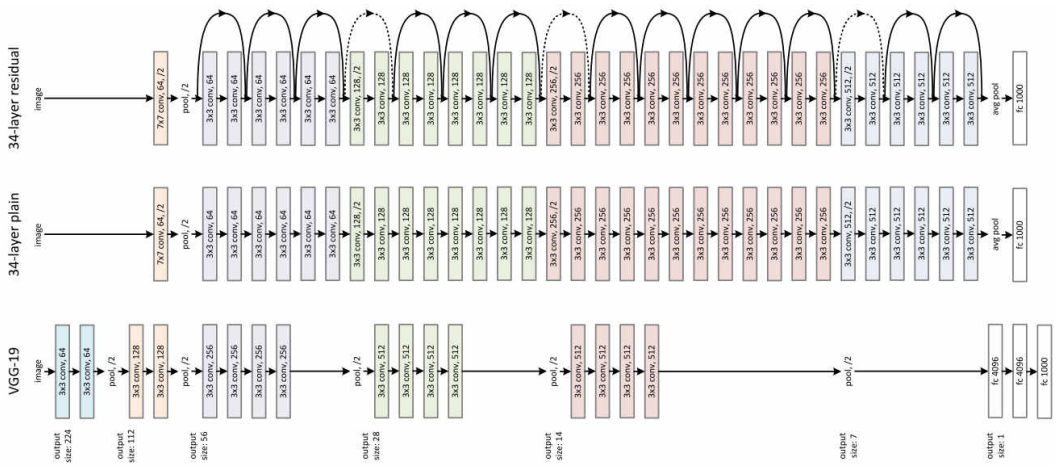


FIGURE 2.34 – *Réseau ResNet*

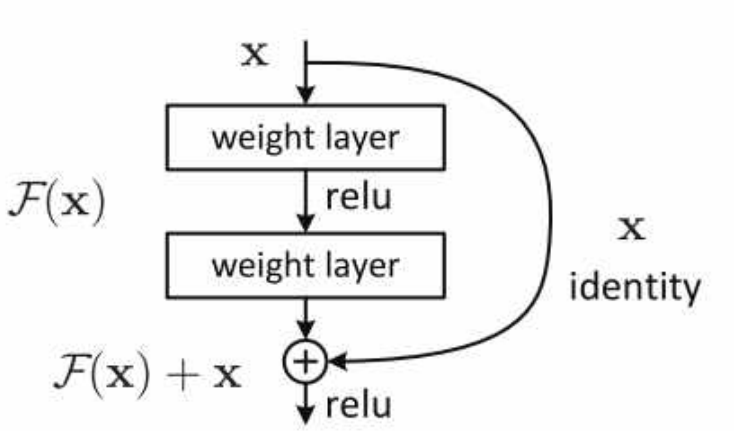


FIGURE 2.35 – *Residual block - Réseau ResNet*

2.3.3 Segmentation sémantique

Un objet sur une image est représenté par des pixels. La segmentation consiste à assigner chaque pixel de cette image à une classe *Figure 2.36*. La segmentation sémantique est utilisée dans la vie courante pour localiser des objets dans une image et dans le secteur médical permettant par exemple de localiser avec précision le disque lombaire et son apophyse afin d'aider les médecins à diagnostiquer le type d'hernie discale dont souffre un patient [5].

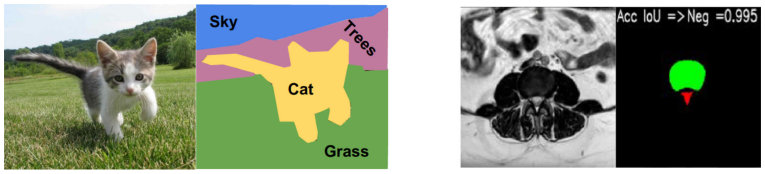


FIGURE 2.36 – *Exemples de segmentation - À gauche, segmentation d'un chat et de son environnement - À droite, segmentation du disque vertébral Mbarki et al. [5]*

Plusieurs types d'architecture par apprentissage profond sont proposés dans la littérature pour réaliser la segmentation sémantique [80].

Le premier réseau de segmentation a été proposé par Log et al. [81]. Il est basé sur un réseau convolutif ne possédant pas de couches totalement connectées afin de se libérer de la taille de l'image d'entrée. Le réseau est inspiré de l'architecture VGG16 en y amputant les couches totalement connectées *Figure 2.37*.

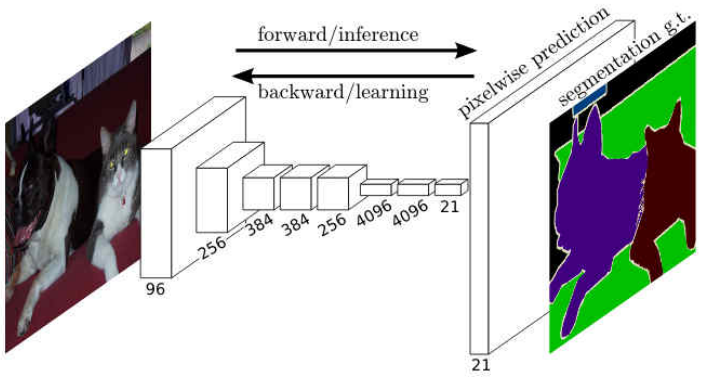


FIGURE 2.37 – *Architecture du premier réseau de segmentation Long et al. 2015*

La dernière opération du réseau est une opération de déconvolution qui permet d'atteindre la définition de l'image d'entrée du réseau. Des déconvolutions à des stades antérieurs sont réalisées et fusionnées pour obtenir les masques de chaque classe *Figure 2.38*. Cette fusion d'informations sur les masques à divers niveau du réseau permet d'augmenter la précision de la segmentation.

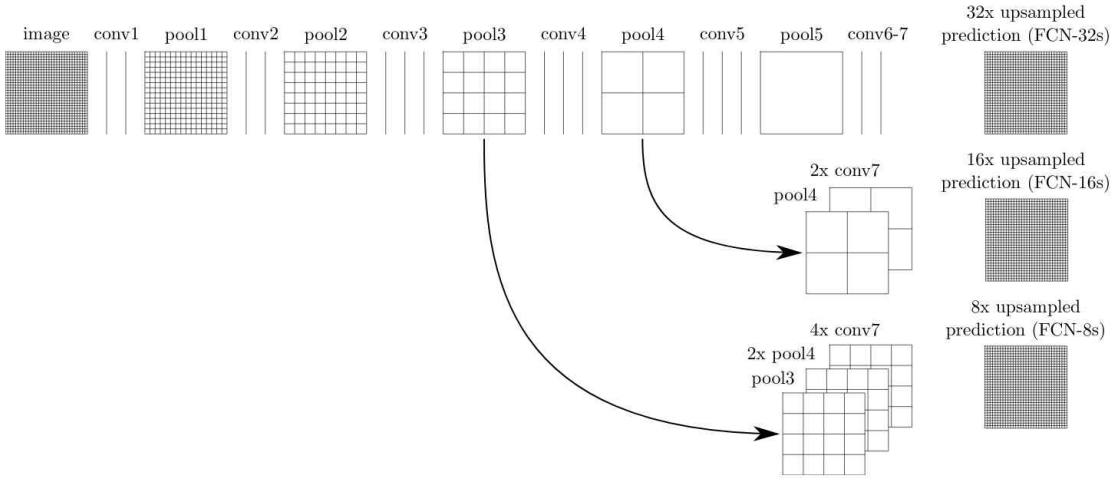


FIGURE 2.38 – *Fusion à divers stade des opérations de déconvolutions Long et al. 2015*

Afin d'augmenter la précision de localisation de l'objet, Chen et al. ont proposé une architecture autour d'un réseau de segmentation sémantique basé sur un réseau convolutif et un réseau CRF (Conditional Random Fields) complètement connectés [82] *Figure 2.39*.

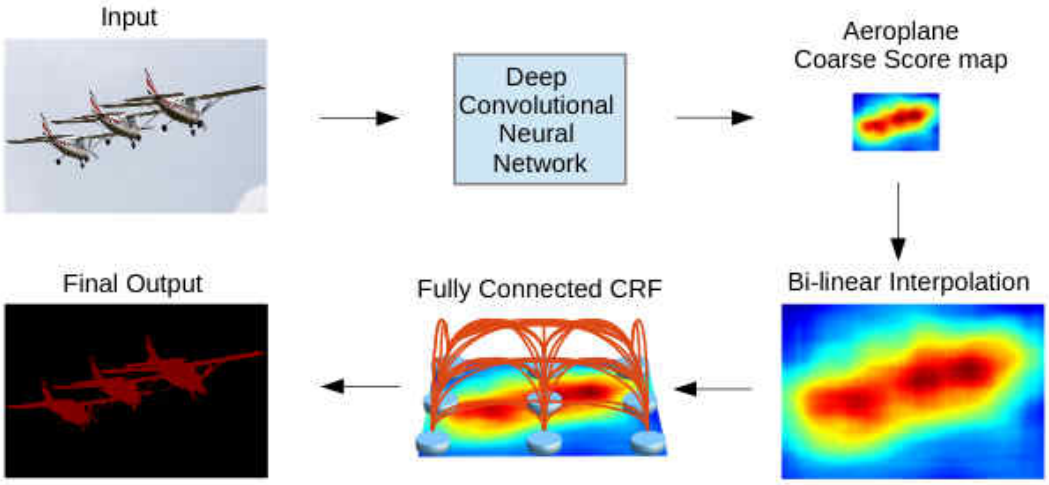


FIGURE 2.39 – *Réseau CNN + CRF - Le masque issu du réseau convolutif (CNN) subit une interpolation qui est ensuite envoyée sur un CRF afin d'affiner la segmentation.*

Une autre famille de réseau de segmentation appelé codage-décodage (Encoder decoder model) ont émergés à partir de 2015. Ces types de réseaux possèdent deux chemins. Le premier chemin composé d'opérations de convolution et de pooling permet de coder les informations issues de l'image d'entrée. La taille de l'image est fortement réduite. Enfin, le second chemin utilise des opérations de déconvolution afin d'augmenter la taille de l'image et de définir les masques des classes recherchées. Cette opération s'apparente à du décodage d'informations. Noh et al. [6] ont été les premiers à publier un article sur une architecture de réseau de segmentation basée sur ce type de structure *Figure 2.40*. Le chemin de codage est échafaudé autour d'une structure VGG16. Le réseau de déconvolution est constitué d'opérations de déconvolution et de unpooling permettant d'identifier la classe d'appartenance de chaque pixel de l'image. Ce réseau a obtenu un score de 72,5% sur la moyenne des intersections over union (IoU) des classes de la base de données PASCAL VOC 2012 segmentation.

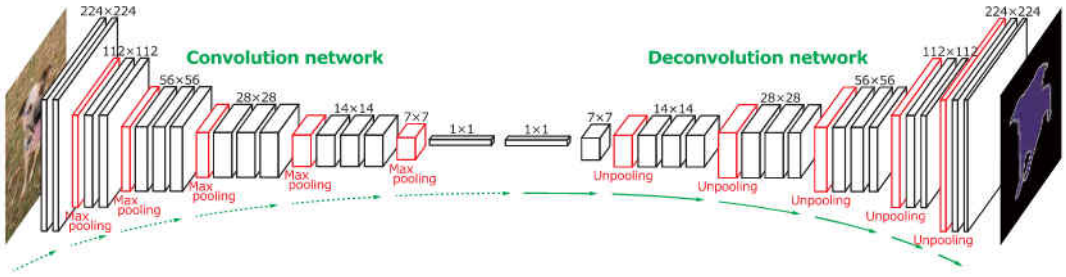


FIGURE 2.40 – *Réseau de codage et de décodage Noh et al. [6]*

Badrinarayanan et al. [7] ont proposé en 2017 une architecture Encodeur-décodeur nommée Segnet dont le chemin décodage est similaire à Noh et al. et basée sur une ar-

chitecture VGG16 *Figure 2.41*. Le chemin de décodage qui est une symétrie du chemin de codage possédant le même nombre de convolutions et de blocs. Chaque couche du codage est liée à une couche de décodage. Un sur-échantillonage (unpooling) est appliqué afin d'augmenter la taille des cartes de caractéristiques pour atteindre la taille de l'image d'entrée. L'originalité provient du fait que les indices des opérations de max-pooling sont mémorisés et réinjectés dans les opérations de unpooling du décodeur. Cette technique permet de diminuer la perte de la résolution spatiale des objets à segmenter. L'opération de unpooling de par sa nature engendre des données éparses sur les cartes de caractéristiques. Les opérations de convolutions du chemin de décodage ont pour objectif de les densifier. La probabilité d'appartenance à une classe est obtenue par un softmax en fin de réseau.

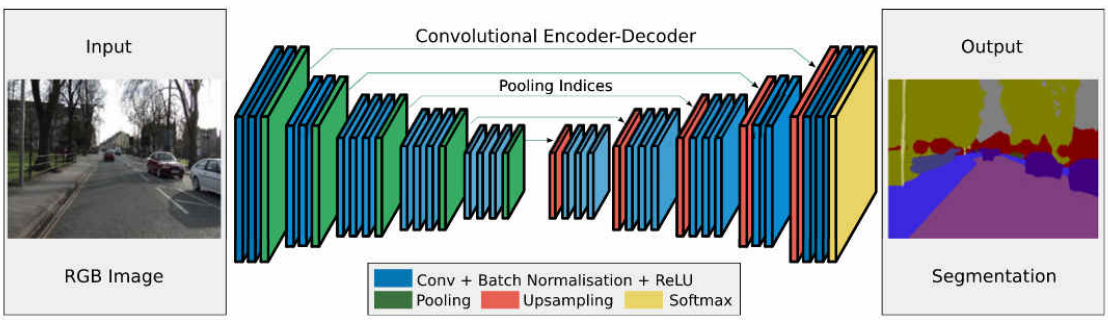


FIGURE 2.41 – *Architecture du réseau Segnet Badrinarayanan et al. [7]*

De multiples variantes utilisant ce principe de codeur-décodeur mettant en oeuvre des opérations de déconvolution ont été proposées [83], [84] et [85]. La principale faiblesse de ces architectures reste dans la perte de finesse dans les masques générés par l'apprentissage.

Ronneberger et al proposent en 2015 un modèle destiné à la biologie et plus particulièrement à l'imagerie microscopique [86]. L'architecture du réseau proposé possède une structure encodeur-décodeur *Figure 2.42*. Le codeur est composé de convolutions avec des noyaux 3x3 et des max-pooling. Le décodeur quant à lui est une succession de déconvolutions et de convolutions, pour créer à la suite d'une convolution de 1x1 les masques. La particularité du réseau est la copie des cartes de caractéristiques du codeur dans le chemin de décodage afin de ne pas perdre les informations de localisation et améliorer la qualité de la segmentation.

Un modèle destiné à l'imagerie médicale, basé sur une architecture proche de U-Net, est proposé en 2016 par Milletari et al.[87]. Cet algorithme permet une segmentation en 3 dimensions.

Une autre technique bien connue en segmentation est le réseaux convolutif basé sur des régions (R-CNN). He et al. en 2017 [8] proposent un réseau convolutif qui dans un premier temps met en place des fenêtres d'intérêts (bouding box) pouvant contenir le ou les objets à segmenter. Le R-CNN est composé de 3 branches de sortie : la première recherche les fenêtres d'intérêts , la seconde sélectionne et associe les fenêtres aux classes à détecter et la dernière réalise le masque dans la fenêtre sélectionnée.

De multiples autres architectures inspirées des structures et des algorithmes présentés

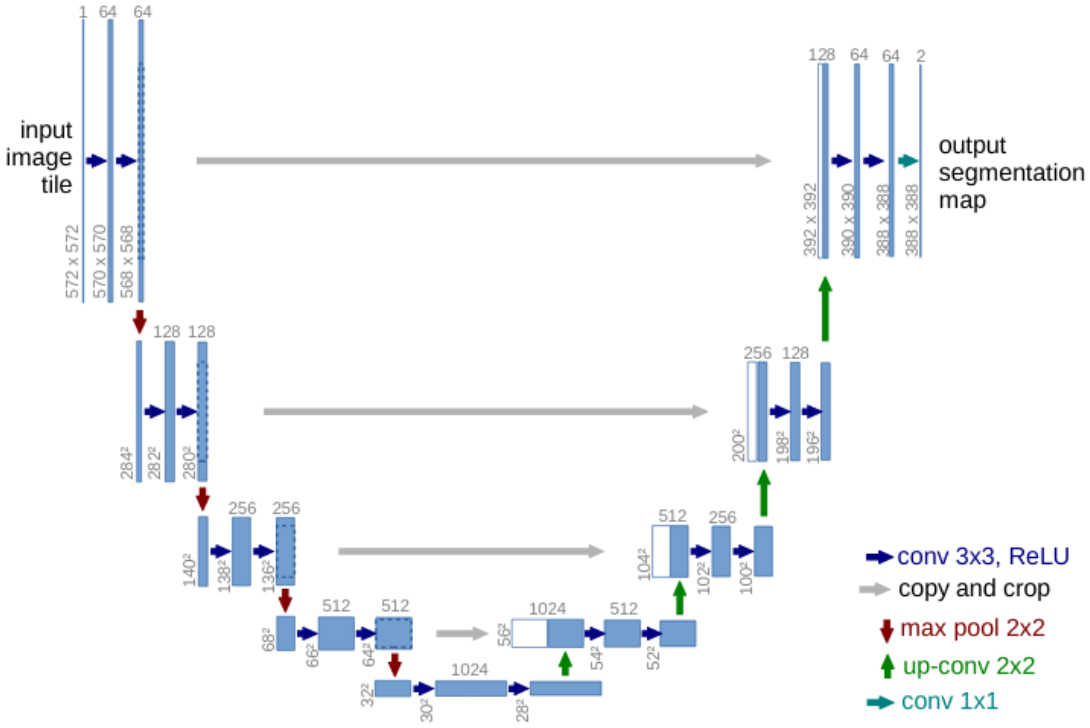


FIGURE 2.42 – *Architecture du réseau U-Net*

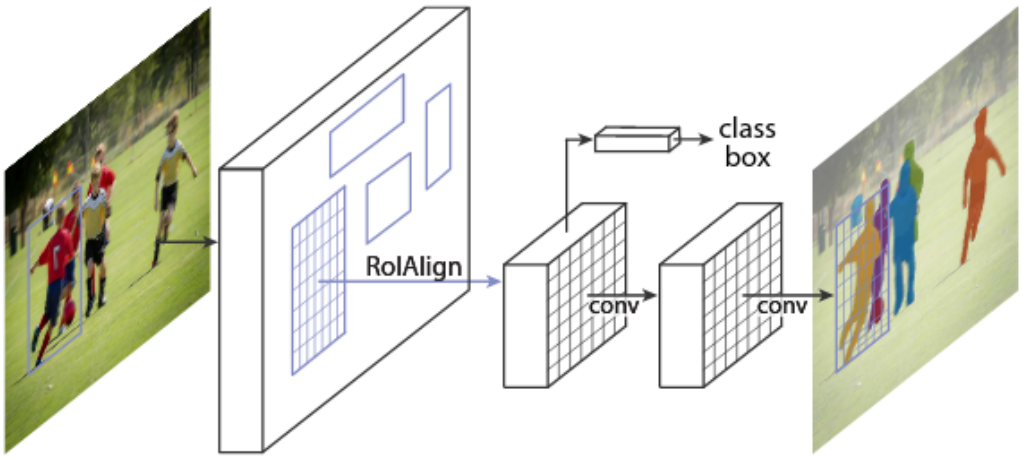


FIGURE 2.43 – *Fonctionnement du R-CNN pour la segmentation d'objets - He et al. en 2017 [8]*

ont vu le jour dans le cadre de la segmentation d'objet [80], [88], [89]. Les convolutions à trou ("atrou convolution") ont été exploitées comme dans les générations des deeplab permettant d'étendre le champ réceptif des opérations de convolution sans alourdir le temps de calcul d'obtention des masques [90], [91] et [92].

2.3.4 Apprentissage

La phase d'apprentissage revêt une importance majeure pour une bonne généralisation lors de l'utilisation du réseau sur des images inconnues. Elle est fortement liée à la qualité de la base d'apprentissage et des opérations de pré-processing réalisées sur celle-ci. Les réseaux profonds possèdent des structures, la plupart du temps lourdes possédant un nombre de paramètres d'apprentissage très important. De ce fait, cette phase demande un grand nombre de données afin d'éviter le surapprentissage et la perte de la capacité de généralisation du réseau . Afin d'accroître artificiellement la base d'apprentissage, l'utilisation à des techniques d'augmentation de données est généralement utilisée.

Data pré-processing

Centrage sur la moyenne

Les données, et en particulier les images, peuvent avoir des variabilités importantes dans les valeurs des pixels. Ceux-ci peuvent, sous 8 bits de codage, varier entre 0 et 255 pour chaque canal. La plupart du temps, il est effectué un recentrage du nuage de point afin d'obtenir une variabilité dans toutes les dimensions de l'espace. Cette opération consiste à retrancher la moyenne de toutes les données à chaque donnée. Dans le cas des images RGB où contenant plusieurs canaux, il est possible de réaliser ce recentrage sur la moyenne soit en calculant la moyenne de chaque canal ou de la moyenne de tous les canaux confondus.

$$\overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ ce qui donne les nouvelles données centrées } x'_i = x_i - \overline{x}$$

Normalisation

La normalisation a pour objectif de recentrer le nuage de point autour de zéro en réduisant sa dispersion. Pour ce faire chaque donnée subit un centrage sur la moyenne *Figure 2.44*. La normalisation est utile si les données possèdent une grande variabilité.

$$\overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ ce qui donne } x'_i = \frac{x_i - \bar{x}}{x_{max} - x_{min}}$$

Une méthode, plus simple consiste à estimer que les pixels d'une image sont codés sur 8 bits et varient donc entre 0 et 255. Il est possible de retrancher 127. Cette méthode transforme les données dans une plage de valeurs comprise entre -1 et +1 sans lire toutes les données et sans calculer la moyenne des données d'entraînement. Ceci évite également de recopier cette moyenne dans les fichiers d'évaluation sur des images inconnues.

$$x'_i = \frac{x_i - 127}{255} \text{ plage de valeurs [-0.5 ;0.5]}$$

$$x'_i = \frac{x_i - 127}{127} \text{ plage de valeurs [-1 ;1]}$$

Standardisation

La standardisation est fortement utilisée dans l'apprentissage profond. Elle consiste à centrer les données autour de la moyenne avec un écart type unitaire *Figure 2.44*. Cette opération peut se réaliser sur l'ensemble des données d'apprentissage ou par mini-paquets.

$$\overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ et } \sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \overline{x})^2}$$

$$x'_i = \frac{x_i - \overline{x}}{\sigma_x}$$

m étant le nombre d'échantillons de la base d'entraînement ou le nombre de données dans les mini-paquets.

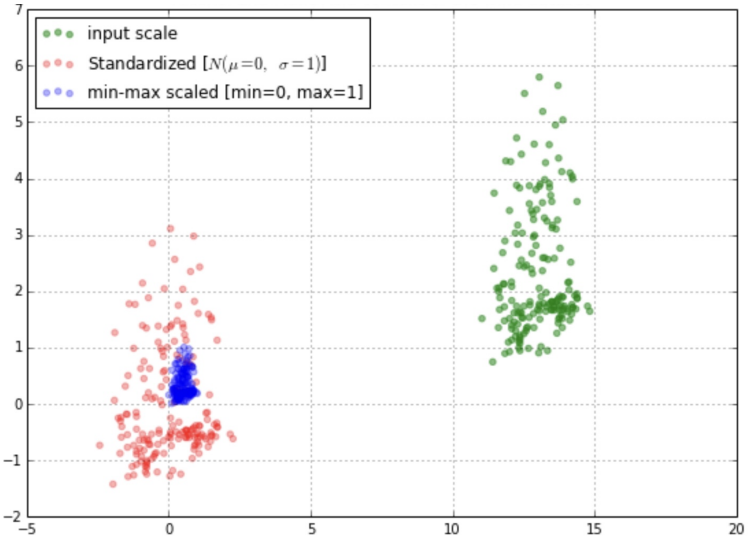


FIGURE 2.44 – *Effet de la normalisation et la standardisation des données - 2 dimensions*

Augmentation des données

La qualité et la quantité de données est un élément important, voire primordial dans l'aboutissement de l'entraînement d'un réseau et de son pouvoir de généralisation. Si la taille de la base de données n'est pas suffisante par rapport aux paramètres du réseau à ajuster, le réseau risque le surentraînement. Des objets peuvent apparaître dans des orientations, des échelles, en totalité ou partiellement masquées, avec des luminosités variables et des teintes différentes. Les bases de données d'objets peuvent manquer d'image pour réaliser l'entraînement du réseau. Dans ce cas, des techniques d'augmentation d'images peuvent être mises en place [93] et [94]. Les transformations appliquées sur les images peuvent être de nature simplement géométrique, chromatique ou d'ajout de bruit... Une combinaison de plusieurs transformations est possible. Lors de la phase d'apprentissage, les images d'entrées possèdent une taille fixée. Après transformation, les images doivent subir une remise à l'échelle pour atteindre la taille des images de la base de données.

Transformation géométriques

Les transformations géométriques classiques sont au nombre de 4 :

- Flipping : consiste à réaliser un effet miroir vertical ou horizontal sur l'image *Figure 2.45*.
- Rotation : Réalisation d'une rotation de l'image engendrant une rotation de l'objet à classifier ou segmenter *Figure 2.45*.
- cropping : Réduction de la taille de l'image.
- Translation : Décalage de l'image vers la droite ou la gauche et/ou vers le haut ou le bas. Cette technique s'accompagne d'une réduction de l'image par la sélection d'une fenêtre qui se déplace dans l'image *Figure 2.45*.

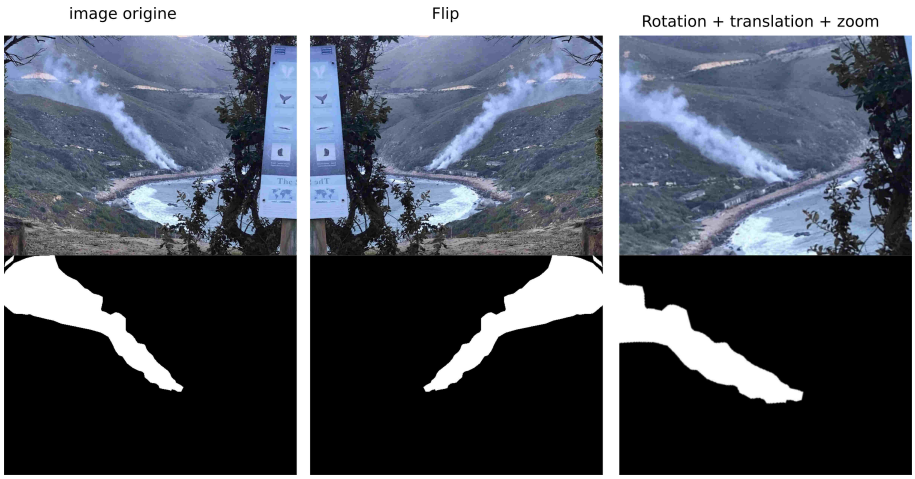


FIGURE 2.45 – *Exemples de transformations géométriques avec les masques de fumées correspondants*

Remarque : dans le cas d'une base de données de segmentation contenant des masques de segmentation, les transformations géométriques doivent être réalisées à l'identique sur les images et les masques.

Transformations non géométriques

- Noise injection : Cette transformation consiste à injecter sur l'image d'origine du bruit Gaussien à partir d'une matrice de données aléatoire. Ajouter du bruit à l'image augmente la robustesse de la détection ou segmentation d'un réseau convolutif [95].
- Color transformation : la plupart des images sont codées sur 3 canaux RGB. La transformation colorimétrique consiste à manipuler les histogrammes de chaque canal en modifiant les plages maximales et minimales ou en appliquant des filtres. Cette méthode peut être source d'erreur dans le cas ou la couleur revêt une information importante pour la détection de certains objets *Figure 2.46*.
- Kernel filter : Cette technique consiste à applique un filtre prenant la forme d'une matrice de taille $n \times n$ sur l'image afin de réaliser un effet de floue ou de renforcement de la netteté. Kang et al. [10] utilisent un filtrage avec un noyau de taille $n \times n$ qui

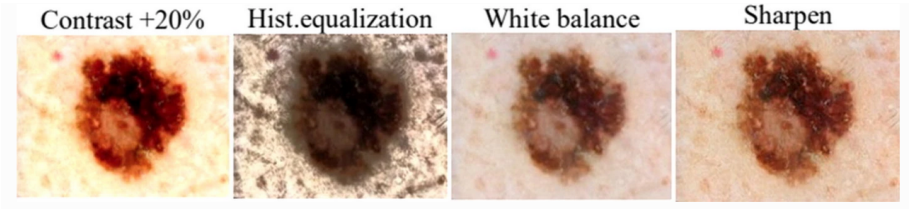


FIGURE 2.46 – *Exemples d'augmentation colorimétrique par Mikolajczyk and Grochowski [9] dans le domaine de la classification des mélanomes*

mélange aléatoirement des données de l'image *Figure 2.47*.

FIGURE 2.47 – *Exemple de Kernel transformation de Kang et al. [10] pour différentes tailles du filtre*

- Random erasing : Cette technique peut être vue comme un dropout de l'image. Une partie de l'image sélectionnée aléatoirement subit une occlusion [11]. Elle permet de prévenir le surentraînement en altérant une partie de l'image et force le réseau à trouver de nouvelles caractéristiques descriptives de l'objet à classifier.

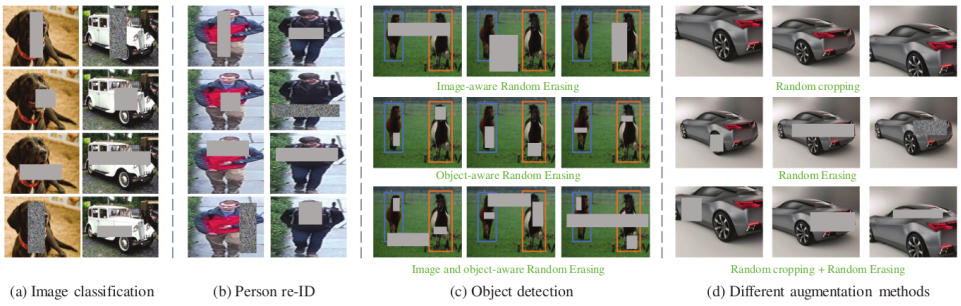


FIGURE 2.48 – *Exemples d'Erasing transformation de Zhong et al. [11]*

Générateur de modèle

Cette méthode consiste à générer de nouvelles images à partir de modèles de la base d'entraînement. Le modèle génératif le plus connu est le GAN (Generative Adversarial Network) [96] issu de l'apprentissage non supervisé. Il est composé de deux réseaux en compétition : le générateur qui crée des échantillons (images par exemple) et le discriminateur qui tente de déterminer si l'image est réelle ou générée *Figure 2.49*. Ce type de réseau donne de très bons résultats, mais leur convergence n'est pas toujours assurée.

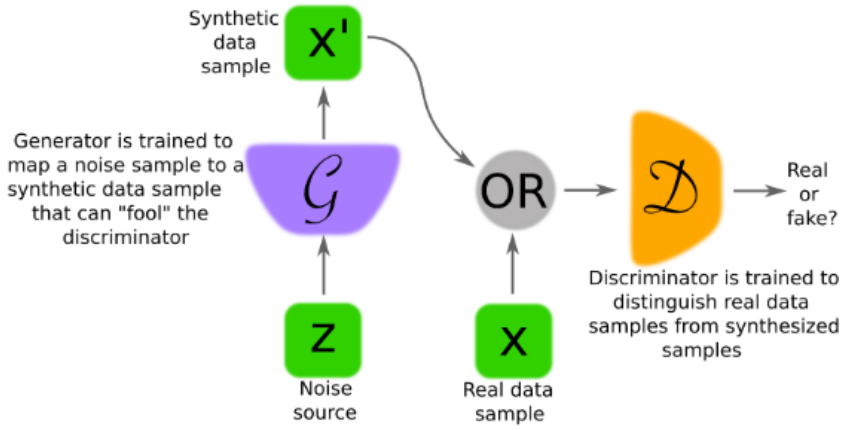


FIGURE 2.49 – *Schéma d'architecture d'un GAN [12]*

Optimiseur

La fonction de perte Loss a pour objectif de définir sur un ensemble des données l'erreur commise entre la prédiction et la réalité. Plus la valeur de cette fonction sera petite et meilleure sera la modélisation du problème sur les jeux de données présentés au réseau. Nous nous plaçons dans le cas d'un apprentissage supervisé. **t** est la valeur réelle et **y** est la valeur prédite. Cette fonction Loss sur l'ensemble des données d'apprentissage se nomme la fonction coût ou cost. Elle peut prendre plusieurs formes comme une valeur absolue

$$Cost = \frac{1}{N}\sum_{i=1}^{N} \mid y_i - t_i \mid$$

ou le carré des erreurs.

$$Cost = \frac{1}{2N}\sum_{i=1}^{N} (y_i - t_i)^2$$

Ces formes sont utilisées plutôt pour des problèmes de régression.

Dans le cadre d'une appartenance à une classe ou non, dans un problème binaire, la fonction coût utilisée la plus couramment utilisée est la fonction régression logistique :

$$Cost = -\frac{1}{N}\sum_{i=1}^{N} t_i Log(y_i) + (1 - t_i) Log(1 - y_i)$$

Cette fonction du fait de la courbe logarithmique pénalise fortement la fonction coût lorsque t=1 et la prédiction proche de 0 ou lorsque t=0 et la prédiction proche de 1.

Si le problème possède plus de deux classes, la fonction d'entropie croisée ou cross entropy est largement utilisée. Cette fonction est couramment utilisée après une fonction softmax qui transforme les résultats d'une classification en probabilité d'appartenance à une classe. La somme de ces probabilité est égale à l'unité. Soit K le nombre de classes du

problème, la fonction entropie croisée (softmax cross entropy with logit) peut s'écrire :

$$Cost = -\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik} Log(y_{ik})$$

Pour une donnée d'entrainement du réseau appartenant à la classe k $t_{ik} = [0, ..., 1, 0, ..]$ avec un vecteur égal à zéro partout sauf dans la classe k (kème position du vecteur).

Dans le cas d'une classification, la fonction coût donne une indication sur l'erreur globale de classification commise par le réseau. L'objectif est d'ajuster les poids du réseau afin de minimiser cette erreur et atteindre une minimum. L'entraînement parfait pourrait atteindre le minimum global. Malheureusement, la plupart des fonctions modélisées sont complexes et non convexes ce qui entraîne l'apprentissage vers un minimum local plus ou moins éloigné du minimum global.

Il existe plusieurs techniques d'optimisation permettant de se rapprocher le plus possible de ce minimum global.

La descente de gradient est la plus connue et la plus simple. C'est un algorithme itératif qui va ajuster les paramètres du réseau afin de se déplacer dans le sens descendant de la pente de la fonction recherchée. On recherche le creux de la fonction qui correspond à un minimum (global ou local). La fonction coût ainsi que les fonctions d'activations utilisées dans le réseau de neurones doivent être dérivables afin de déterminer la pente de la fonction. Dans le cas d'un réseau convolutif, les images d'entrée sont présentées au réseau. Celui-ci va fixer une classe à chaque image. Il suffit de calculer la fonction coût, puis de rétropropager le gradient en sens inverse couche après couche et d'ajuster les paramètres θ_j afin de réduire l'erreur commise et donc la fonction coût. L'ajustement des paramètres est modulé par un pas d'apprentissage λ qui est un hyperparmètre du réseau pouvant être fixe ou variable durant l'apprentissage.

$$\theta_{j,t+1} = \theta_{j,t} - \lambda \frac{\partial Cost(\theta, z)}{\partial \theta_j}$$

L'avantage principal de la descente de gradient reste son défaut. Elle est stable et les courbes de coût sont peu bruitées, mais il est possible de rester bloquer rapidement dans un minimum local.

La descente de gradient stochastique repose sur l'hypothèse que si les variables sont identiquement distribuées, la fonction coût correspond à la moyenne des erreurs de chaque donnée. La mise à jour des paramètres du réseau se fait plus souvent, parfois entre chaque exemple présenté au réseau. La descente de gradient stochastique est plus rapide que la descente de gradient avec une convergence plus lente et engendre beaucoup de bruit. Une variante consiste à créer des mini-paquets à partir des données d'apprentissage préalablement mélangées. On calcule dans ce cas une fonction Loss pour chaque mini-paquet. L'objectif est d'ajuster les paramètres du réseau pour minimiser cette fonction Loss. La taille des mini-paquets est un hyperparamètre à gérer.

$$\theta_{j,t+1} = \theta_{j,t} - \lambda \frac{\partial Loss(\theta, z)}{\partial \theta_j}$$

Avec par exemple (M la taille des mini-paquets)

$$Loss = -\frac{1}{M}\sum_{i=1}^{M} Loss(t_i, y_i) = -\frac{1}{M}\sum_{k=1}^{K}\sum_{i=1}^{M} t_{ik} Log(y_{ik})$$

La sortie du réseau est la plupart du temps envoyée sur une fonction Softmax qui va générer un vecteur ou une matrice contenant la probabilité d'appartenance aux diverses classes *Figure 2.50*. La classe de chaque sortie est attribuée à celle qui possède la plus grande probabilité. Afin de rétropopager le gradient, cette fonction softmax $\sigma(\mathbf{z})$ doit être aisément dérivable :

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

si i=j alors

$$\frac{\partial y_i}{\partial z_i} = \frac{\partial \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}}{\partial z_i} = \frac{e^{z_i}\sum_{j=1}^{K} e^{z_j} - e^{z_i}e^{z_i}}{(\sum_{j=1}^{K} e^{z_j})^2} = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}(1 - \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}) = y_i(1 - y_i)$$

si i $\neq$ j

$$\frac{\partial y_i}{\partial z_i} = \frac{\partial \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}}{\partial z_i} = \frac{0 - e^{z_i}e^{z_j}}{(\sum_{j=1}^{K} e^{z_j})^2} = -\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}\frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_j}} = -y_i y_j$$

Dans le cas ou la sortie de la fonction softmax est envoyée à la fonction cross entropie, la fonction perte pour un exemple $Loss(t_i, y_i)$ avec y_{ik} la probabilité que l'exemple i appartienne à la classe c et t_{ik} est la réponse vraie et est à 1 si l'exemple i appartient à la classe c *Figure 2.50*.

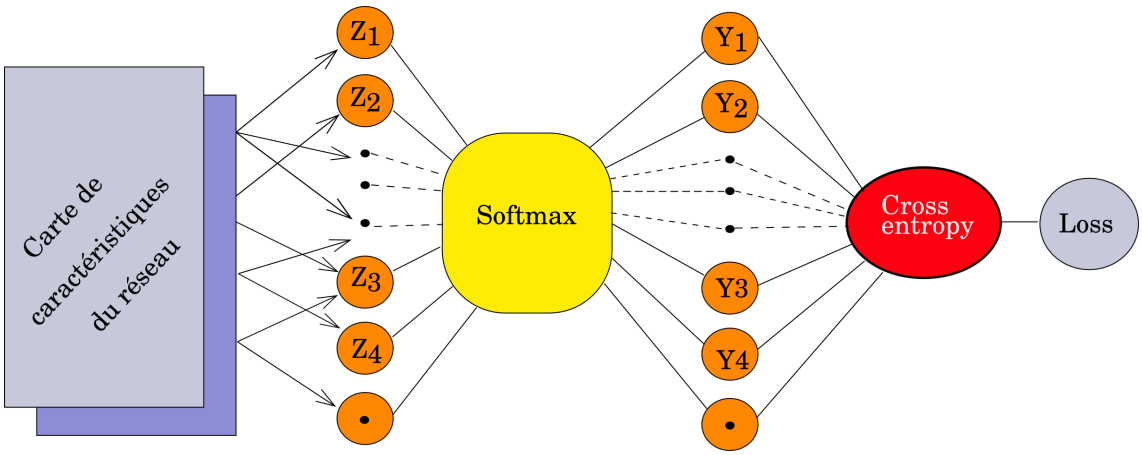


FIGURE 2.50 – *softmax suivi d'une fonction cross entropy*

$$Loss(t_i, y_i) = -\sum_{k=1}^{K} t_{ik} Log(y_{ik})$$

$$\frac{\partial Loss(t_i, y_i)}{\partial z_i} = -\sum_{j=1}^{K} \frac{\partial t_j log(y_j)}{\partial z_i} = -\sum_{j=1}^{K} t_j \frac{\partial log(y_j)}{\partial z_i} = -\sum_{j=1}^{K} t_j \frac{\partial log(y_j)}{\partial y_j} \frac{\partial y_j}{\partial z_i}$$

$$= -\sum_{j=1}^{K} t_j \frac{1}{y_j} \frac{\partial y_j}{\partial z_i} = -\frac{t_i}{y_i} \frac{\partial y_i}{\partial z_i} - \sum_{j \neq i}^{K} \frac{t_j}{y_j} \frac{\partial y_j}{\partial z_i} = -\frac{t_i}{y_i} y_i(1 - y_i) - \sum_{j \neq i}^{K} \frac{t_j}{y_j}(-y_j y_i)$$

$$= -t_i + t_i y_i + \sum_{j \neq i}^{K} t_j y_i = -t_i + \sum_{j=1}^{K} t_j y_i = -t_i + y_i \sum_{j=1}^{K} t_j = y_i - t_i$$

Il reste à réaliser de manière itérative la rétro-propagation du gradient en calculant les dérivations partielles en chaîne sur la fonction Loss par rapport aux paramètres du réseau et par rapport aux sorties des couches successives du réseau *Figure 2.51*. La phase d'initialisation consiste à calculer le gradient de la fonction Loss par rapport à la sortie du réseau (voir calcul plus haut). Les gradients ainsi calculés se rétro-propagent de la sortie vers l'entrée du réseau et permettent d'appliquer une correction sur ses paramètres. La propagation directe calcule les valeurs des sorties des couches de l'entrée vers la sortie avec les paramètres du réseau, la rétro-propagation rétro-propage le gradient de l'erreur et permettent de corriger couche après couche les paramètres du réseau afin de réduire l'erreur entre la prédiction et la réalité sur le jeu de données présenté.

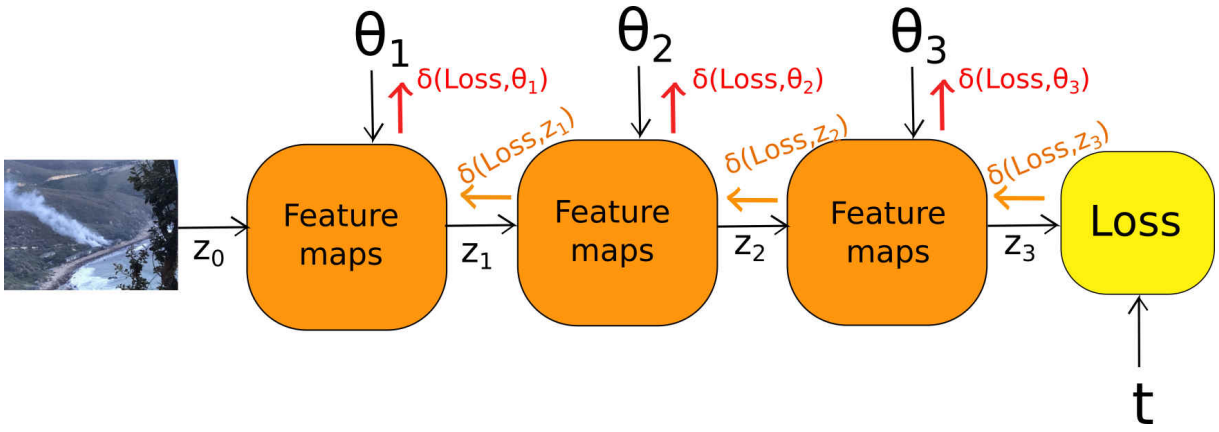


FIGURE 2.51 – *Rétro-propagation du gradient δ par rapport à la sortie précédente ou par rapport aux paramètres du réseau*

Plaçons-nous dans le cadre d'un réseau convolutif. L'algorithme de propagation du gradient étant itératif, il faut pouvoir calculer de manière itérative les dérivées partielles de la fonction Loss par rapport aux sorties des couches successives (cartes de caractéristiques). Les réseaux convolutifs sont composés d'opérations de convolution et de pooling. Afin de calculer le gradient dans un réseau convolutif, prenons un cas simple constitué d'un réseau à une dimension sur deux couches successives.

Convolution

Intéressons-nous tout d'abord à la rétropropagation du gradient dans les opérations de

convolution. La fonction de convolution est une opération linéaire, les dérivées partielles peuvent se calculer simplement. Dans notre exemple, le noyau de convolution se limite à un vecteur $[w_1, w_2, w_3]$ (une dimension). Les données x_i entrent dans la couche h et subissent une opération de convolution pour obtenir les données y_i de la couche h+1 *Figure 2.52*.

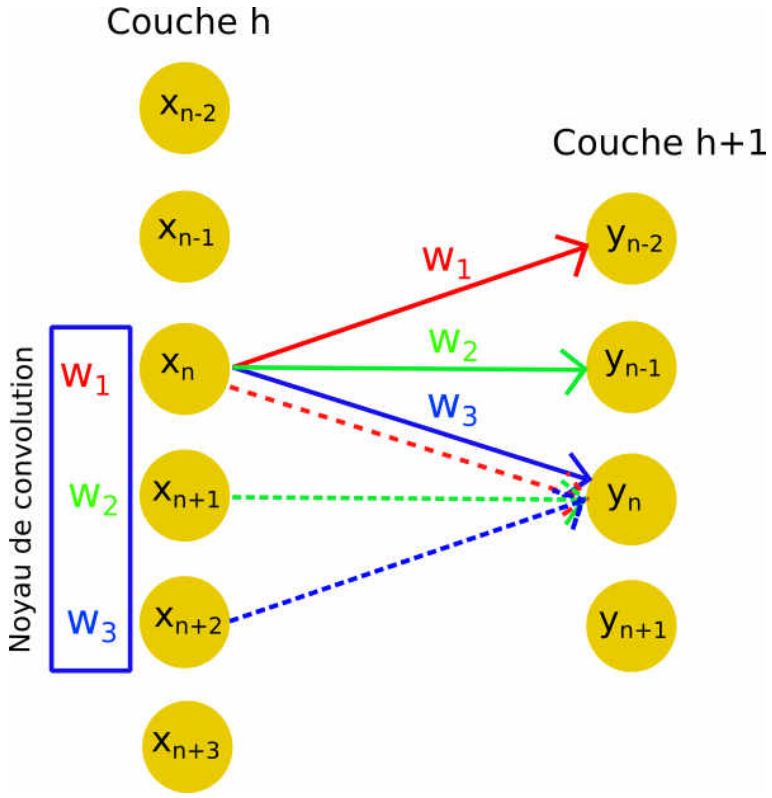


FIGURE 2.52 – *Exemple de convolution - CNN - Rétro-propagation*

L'opération de convolution sans padding (ajout de zéro pour maintenir constante la taille de la carte de caractéristiques après l'opération de convolution) va produire la sortie $y_n = w_1 x_n + w_2 x_{n+1} + w_3 x_{n+2}$. Le gradient de la fonction Loss par rapport à l'entrée x_n s'écrit :

$$\delta_n^x = \frac{\partial Loss}{\partial x_n} = \frac{\partial Loss}{\partial y}\frac{\partial y}{\partial x_n}$$

Le gradient de x_n est fonction des sorties y_{n-2}, y_{n-1} et y_n donc :

$$\delta_n^x = \frac{\partial Loss}{\partial y_{n-2}}\frac{\partial y_{n-2}}{\partial x_n} + \frac{\partial Loss}{\partial y_{n-1}}\frac{\partial y_{n-1}}{\partial x_n} + \frac{\partial Loss}{\partial y_n}\frac{\partial y_n}{\partial x_n}$$

On voit apparaître les gradients des sorties y et des poids du noyau de convolution.

$$\delta_{n-2}^y = \frac{\partial Loss}{\partial y_{n-2}} \; ; \; \delta_{n-1}^y = \frac{\partial Loss}{\partial y_{n-1}} \; ; \; \delta_n^y = \frac{\partial Loss}{\partial y_n}$$

$$w_3 = \frac{\partial y_{n-2}}{\partial x_n} \; ; \; w_2 = \frac{\partial y_{n-1}}{\partial x_n} \; ; \; w_1 = \frac{\partial y_n}{\partial x_n}$$

De manière générale :

$$\delta_n^x = \sum_{i=1}^{W_{size}} \frac{\partial Loss}{\partial y_{n-i+1}} \frac{\partial y_{n-i+1}}{\partial x_n} = \sum_{i=1}^{W_{size}} \delta_{n-i+1}^{(y)} \times w_i$$

Cette opération peut être transformée par une convolution entre le gradient de la sortie y et le flip du noyau de convolution.

$$\delta^x = \delta^y * flip(w)$$

Cette première étape permet de calculer le gradient δ^x connaissant δ^y. Il suffit de réaliser cette opération de couche en couche pour toutes les opérations de convolution.

La seconde étape demande de calculer le gradient Loss par rapport aux poids du noyau de convolution.

$$\frac{\partial Loss}{\partial w_i} = \frac{\partial Loss}{\partial y} \frac{\partial y}{\partial w_i}$$

Notre exemple montre que

$$\begin{cases} y_{n-2} = x_{n-2}w_1 + x_{n-1}w_2 + x_n w_3 \\ y_{n-1} = x_{n-1}w_1 + x_n w_2 + x_{n+1} w_3 \\ y_n = x_n w_1 + x_{n+1} w_2 + x_{n+2} w_3 \\ y_{n+1} = x_{n+1} w_1 + x_{n+2} w_2 + x_{n+3} w_3 \end{cases}$$

Calculons

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial y} \frac{\partial y}{\partial w_1}$$

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial y_{n-2}} \frac{\partial y_{n-2}}{\partial w_1} + \frac{\partial Loss}{\partial y_{n-1}} \frac{\partial y_{n-1}}{\partial w_1} + \frac{\partial Loss}{\partial y_n} \frac{\partial y_n}{\partial w_1} + \frac{\partial Loss}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial w_1}$$

On remarque :

$$\frac{\partial Loss}{\partial w_1} = \delta_{n-2}^y \times x_{n-2} + \delta_{n-1}^y \times x_{n-1} + \delta_n^y \times x_n + \delta_{n+1}^y \times x_{n+1}$$

La même démarche peut être réalisée pour w_2 et w_3. On en déduit la relation générale suivante :

$$\frac{\partial Loss}{\partial w_i} = \frac{\partial Loss}{\partial y} \frac{\partial y}{\partial w_i} = \sum_{n=1}^{x_{size}-w_{size}+1} \frac{\partial Loss}{\partial y_n} \frac{\partial y_n}{\partial w_i}$$

Le gradient de la fonction Loss en fonction des poids du noyau se représente par une opération de convolution.

$$\frac{\partial Loss}{\partial w} = \delta^y * x = x * \delta^y$$

Pooling

La fonction de pooling n'est pas une opération mathématique, mais une réduction spatiale

de la taille de la représentation des données. Cette opération permet de réduire le temps de calcul et surtout d'introduire une invariance à la translation des objets à détecter. Dans la phase de propagation directe, la fonction pooling renvoie un résultat unique d'une fenêtre de taille NxN. La valeur renvoyée dépend du type de pooling. Dans le cas d'un max-pooling, c'est la plus grande valeur de la fenêtre de pooling qui est renvoyée. Pour un mean-pooling, c'est la valeur moyennes des valeurs contenue dans la fenêtre de pooling *Figure 2.53*. Pour la phase de rétro-propagation, la stratégie de rétro-propagation du gradient est la suivante :

- Max-pooling : l'erreur provient uniquement de la valeur ayant été sélectionnée (la plus grande valeur de la fenêtre de pooling). Le gradient uniquement de la cellule gagnante est transmis à la couche précédente, les autres valeurs de cette fenêtre sont neutralisées, car elles ne participent plus à sortie du réseau.
- Mean-pooling : chaque valeur de la fenêtre de pooling va contribuer à la rétro-propagation du gradient avec un coefficient de $\frac{1}{NxN}$.

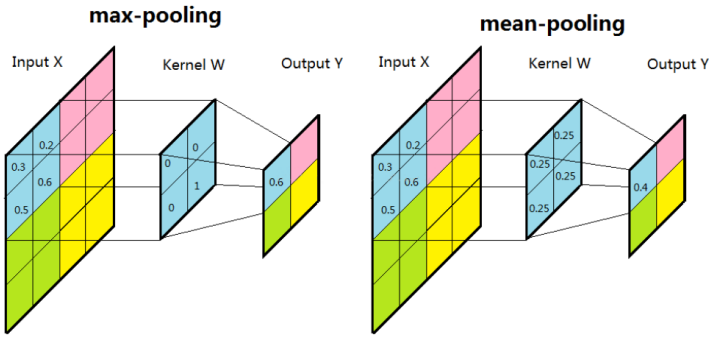


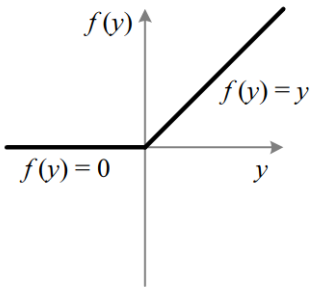
FIGURE 2.53 – *Opération de Pooling -Nan Cui [13]*

Fonctions d'activation

La sortie des opérations de pooling ou de convolution est la plupart du temps envoyée sur une fonction d'activation permettant de moduler la sortie des neurones.

Une fonction d'activation fortement utilisée dans les réseaux convolutifs est la fonction ReLu (Rectified Linear Unit) [97]. Cette fonction demande moins de puissance de calcul que la fonction sigmoïde ou tanh, ce qui explique son utilisation accrue dans le deep learning. Cependant, le fait que les entrées de valeurs négatives de la fonction d'activation deviennent nulles en sortie peut provoquer une perte du gradient pendant la phase d'entraînement du réseau.

$$y_i = \begin{cases} z & si \quad z > 0 \\ 0 & si \quad z \leq 0 \end{cases}$$

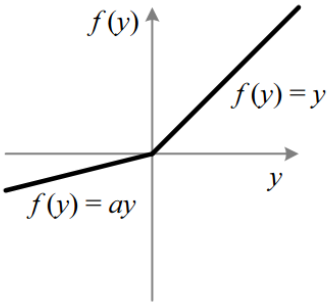


Cette fonction d'activation est dérivable par partie.

$$\frac{\partial ReLu}{\partial y_i} = \begin{cases} 1 & si \quad z > 0 \\ 0 & si \quad z < 0 \end{cases}$$

Une variante de la fonction d'activation ReLu est la fonction Leaky ReLu [98]. Elle possède une pente faible pour les valeurs négatives qui permet de restreindre la perte de gradient et la mort de certaines cellules.

$$y_i = \begin{cases} z & si \quad z > 0 \\ \alpha z & si \quad z \leq 0 \end{cases}$$



Cette fonction d'activation est dérivable par partie.

$$\frac{\partial leakyReLu}{\partial y_i} = \begin{cases} 1 & si \quad z > 0 \\ \alpha & si \quad z < 0 \end{cases}$$

Dans le cadre d'un réseau convolutif, le gradient se rétro-propage d'opération en opération permettant l'ajustement des paramètres du réseau dans la direction opposée du gradient. Des librairies sous python et C++ tel que Tensorflow, Theano, caffe, torch, Keras, etc. permettent de réaliser itérativement cette rétro-propagation du gradient et la mise à jour des paramètres du réseau permettant de se concentrer sur la structure de celui-ci.

Amélioration de la descente de Gradient

Momentum

Il est possible d'améliorer l'algorithme de descente de gradient en ajoutant le moment de la dynamique des paramètres d'apprentissage. Ce moment est analogue à une dynamique Newtoniene car il s'agit d'ajouter un historique de la variation des paramètres assimilable à la vitesse donnant une indication sur les directions antérieures prises. Un hyperparamètre α est ajouté qui joue le rôle de paramètre d'oublie des variations antérieures. Si $\alpha = 0$, on se retrouve dans un algorithme simple de descente de gradient. Plus α est grand et plus l'histoire des modifications antérieures des paramètres a un poids un peu comme une inertie. L'ajustement des paramètres du réseau devient :

$$v_{j,t+1} = \alpha v_{j,t} - \lambda \frac{\partial Loss(\theta, z)}{\partial \theta_j}$$

et

$$\theta_{j,t+1} = \theta_{j,t} + v_{j,t+1}$$

Nesterov momentum

Le Nesterov momentum est une méthode d'optimisation permettant d'accélérer la phase d'apprentissage. La différence entre l'optimisation par momentum réside dans le calcul du gradient. Dans cette méthode d'optimisation, le gradient n'est pas calculé avec des paramètres θ courants, mais des paramètres intermédiaires $\widetilde{\theta}$ avec :

$$\widetilde{\theta}_j = \theta_{j,t} + \alpha v_{j,t}$$

Le calcul du gradient avec les paramètres intermédiaires :

$$g_j = \frac{\partial Loss(\theta, z)}{\partial \widetilde{\theta}_j}$$

$$v_{j,t+1} = \alpha v_{j,t} - \lambda g_j$$

Avec la mise à jour des paramètres du réseau.

$$\theta_{j,t+1} = \theta_{j,t} + v_{j,t+1}$$

Cette méthode permet d'accélérer la recherche du minimum global et réduit les oscillations. Si la vitesse mise à jour oriente le réseau vers une augmentation de la fonction Loss, le gradient en tient compte et corrige les paramètres en conséquence évitant ainsi les oscillations [99]. Les optimiseurs Momentum ou Nesterov momentum demandent l'ajout d'un hyperparamètre à régler manuellement.

Adagrad et RMSProp

L'optimiseur Adagrad possède la particularité de modifier le pas d'apprentissage durant l'apprentissage [100]. Des mises à jours plus importantes seront réalisées pour les exemples moins fréquents.

$$\theta_{j,t+1} = \theta_{j,t} - \frac{\lambda}{\sqrt{G_t + \epsilon}} \frac{\partial Loss(\theta, z)}{\partial \theta_j}$$

Avec G_t est la somme du carré des gradients antérieurs pour tous les paramètres. ϵ est un paramètre évitant la division par zéro de l'ordre de 10^{-8}. Ce type d'optimiseur évite la recherche d'un pas d'apprentissage. Le pas d'apprentissage est adapté aux données présentées au réseau. Le désavantage de cette méthode est une réduction permanente du pas d'apprentissage et un coût de temps calcul important.

L'algorithme RMSProp (Root Mean Squared Prop) est une variante de l'algorithme Adagrad qui possède de meilleures performances sur les fonctions non convexes et converge plus rapidement. La méthode d'Adagrad diminue trop fortement le pas d'apprentissage en tenant compte de l'historique total de l'apprentissage et peut être piégée dans un minimum local. RMSProp divise simplement le taux d'apprentissage par une moyenne en décroissance exponentielle.

L'algorithme Adam (Adaptative moment) [101] est une combinaison des méthodes RMSProp et Momentum. Cette méthode stocke la moyenne décroissante des gradients carrés (RMSProp). Il contient également la moyenne du gradient passé similaire à Momentum.

En conclusion, la méthode d'optimisation doit être choisie au regard de la base de données et de la connaissance de l'algorithme. Pour des données éparses, un optimisateur avec une adaptation du pas d'apprentissage semble le plus adapté. Toutefois, dans son état de l'art [102] sur les algorithmes d'optimisation, Sébastien Ruder conseille d'utiliser en priorité l'algorithme d'Adam. De plus, Kingman [101] compare sa méthode aux autres algorithmes d'optimisation et montre que l'algorithme d'Adam est en mesure de résoudre efficacement les problèmes d'apprentissage profond *Figure 2.54*.

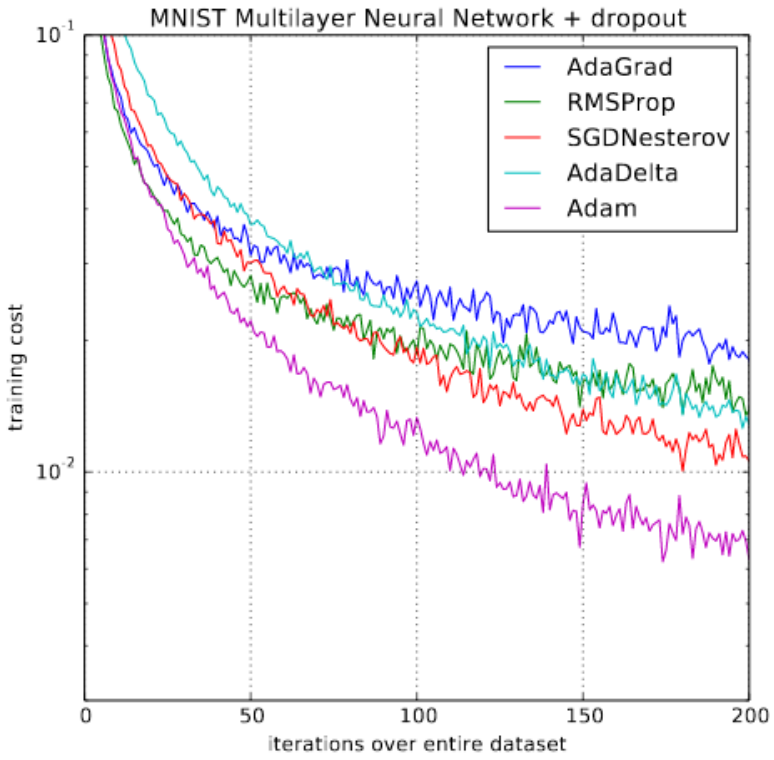


FIGURE 2.54 – *Comparaison des divers algorithmes d'optimisation - Kingman et al 2014*

Chapitre 3: Détection du feu et de la fumée

Contents

La détection du feu et de la fumée, par les méthodes classiques, demandait une expertise afin de déterminer les vecteurs de caractéristiques à prendre en compte pour la classification. L'apprentissage profond supervisé possède comme avantage primordial de se soustraire de cette contrainte. Les paramètres du réseau lors de la phase d'apprentissage vont s'ajuster et trouver une structure dans les données qu'aucun expert humain n'aurait trouvée. La pierre de voûte de ces techniques réside dans la qualité et la quantité de la base de données. Le réseau a besoin de nombreuses données variées afin de classifier convenablement les images. Une bonne généralisation sur des images inconnues va demander une variété d'images de feu et de fumée reflétant la réalité.

Si nous considérons que chaque pixel de l'image de chaque canal représente une donnée d'entrée, dans le cas d'un réseau totalement connecté, le nombre de connexions avec la couche n+1 peut, en fonction de la taille de l'image d'entrée, engendrer un nombre de paramètres très important. Par exemple : une image d'entrée codée sur 3 canaux de taille 640x480 pixels, dans le cas ou il n'y pas de connexion inter couche et pour une première couche de la même taille que l'image d'entrée, demanderait près de 1.10^{12} paramètres à ajuster pour la première couche. Ce réseau basique serait difficile à entraîner et non utilisable en temps réel avec les technologies actuelles.

Les réseaux convolutifs, en partageant les poids du masque de convolution, permettent de réduire fortement les paramètres du réseau. De plus, l'opération de convolution de par son champ réceptif conserve la structure spatiale de l'image dans les diverses cartes de caractéristiques 3.1. De plus, l'information est traitée de la même manière sur l'ensemble de l'image et la disposition spatiale est conservée.

Les opérations de convolution sont la plupart du temps suivies d'opérations de pooling qui réduisent la taille des cartes de caractéristiques. La réduction des cartes de caractéristiques engendre une invariance en translation de l'objet à détecter.

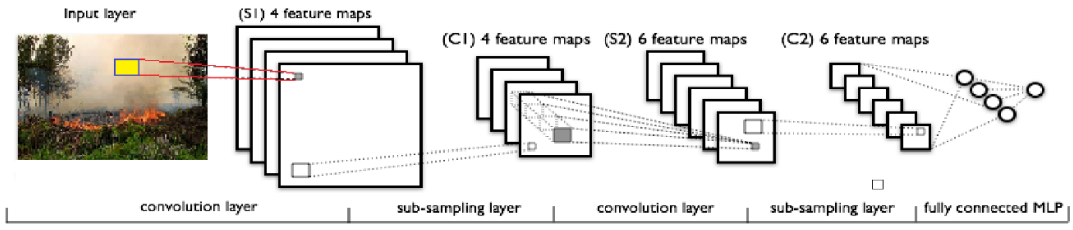


FIGURE 3.1 – *Convolutional Neural Network architecture*

La suite de ce chapitre relate le travail de recherche réalisé dans l'article "Convolutional neural network for video fire and smoke detection" [21].

3.1 Base de données

La base de données est un élément essentiel dans le cadre de l'apprentissage d'un réseau profond. Sa qualité va influencer la faculté du réseau à généraliser la classification en dehors de la base d'apprentissage. L'objectif est de classifier les images en 3 classes :

- feu
- fumée
- background (ni feu, ni fumée)

L'apprentissage est de type supervisé, consistant à présenter des images au réseau en connaissant le label de sortie ou la classe correspondante. La base de données d'images a été réalisée à l'aide d'une application conçue pour l'occasion sous python en utilisant la librairie OpenCv. Sur des images récupérées sur internet, des découpes de fenêtre de taille 64x64 pixels contenant les "objets" feu, fumée et background ont été effectuées *Figure 3.2*.

FIGURE 3.2 – *Exemple d'images 64x64 pixels pour les trois classes*

Afin d'augmenter le nombre d'images de la base, des opérations de flip, rotation, modification de la luminosité ont été réalisées. La répartition des 27 919 images de la base est donnée dans le tableau 3.1.

La base de données est subdivisée en trois sous-ensembles train, valid et test correspondant respectivement à 60%,20% et 20% de la base. La base de validation a pour objectif de

Labels	Train	Valid	test
Feu	4353	1450	1450
Fumée	5350	1782	1782
Background	7052	2350	2350
Total	*16755*	*5582*	*5582*

TABLE 3.1 – *Nombre de données de la base d'images*

vérifier le bon apprentissage du réseau et d'éviter le surapprentissage. La base de test permet à l'issue de l'apprentissage d'évaluer la qualité de la classification sur un sous-ensemble d'image jamais vu par le réseau.

3.2 Architecture du réseau

Le réseau choisi est léger afin de permettre une classification compatible avec du temps réel. Il est constitué de deux parties. La première partie est dédiée au codage de l'information et comporte des opérations de convolutions avec des noyaux 3x3 pixels et d'opération de max-pooling. La seconde partie est composée de deux couches totalement connectées de 100 cellules chacune finissant sur 3 cellules correspondant aux 3 classes.

L'image RGB codée sur 3 canaux entre dans le réseau et subit une première opération de convolution donnant naissance à 16 cartes de caractéristiques de taille de 62x62 pixels. Une seconde opération de convolution, puis un max-pooling de noyau 3x3 et de pas 2 produit 16 cartes de caractéristiques de taille 29x29 pixels. Deux autres opérations de convolution suivie d'un max-pooling amènent à la dernière carte de caractéristique (couche 6) de dimension 12x12 pixels. Cette dernière couche de caractéristiques est liée ensuite à une couche totalement connectée (couche 7). La seconde couche totalement connectée est enfin liée aux 3 sorties du réseau.

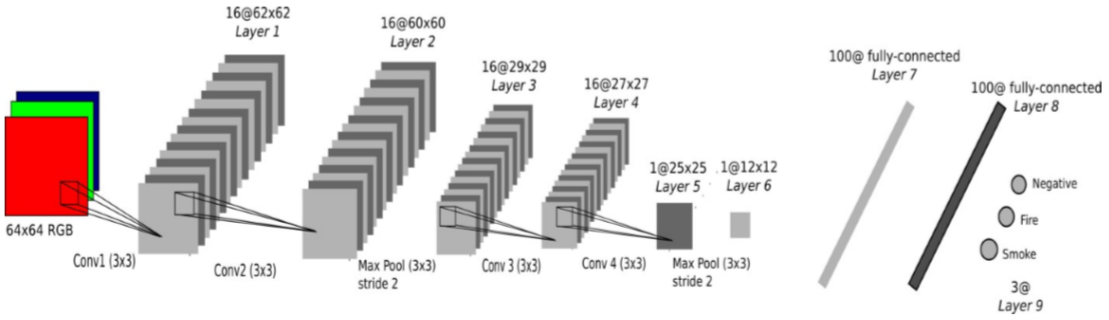


FIGURE 3.3 – *Architecture du réseau*

Nous avons choisi pour les couches totalement connectées et les couches de convolutions une fonction d'activation Leaky ReLu (Rectified Linear Unit) avec un coefficient de $a = \frac{1}{3}$ *Figure 3.4*. La fonction ReLu permet une rapidité de calcul tant dans le calcul direct que dans la phase d'entrainement via la rétropropagation et le calcul de sa dérivée. De plus, Leaky ReLu possède une pente faible dans les valeurs négatives et permet d'éviter la saturation des neurones et la perte du gradient.

La sortie du réseau utilise une fonction Softmax *Equation 3.1* avec une distribution sur 3

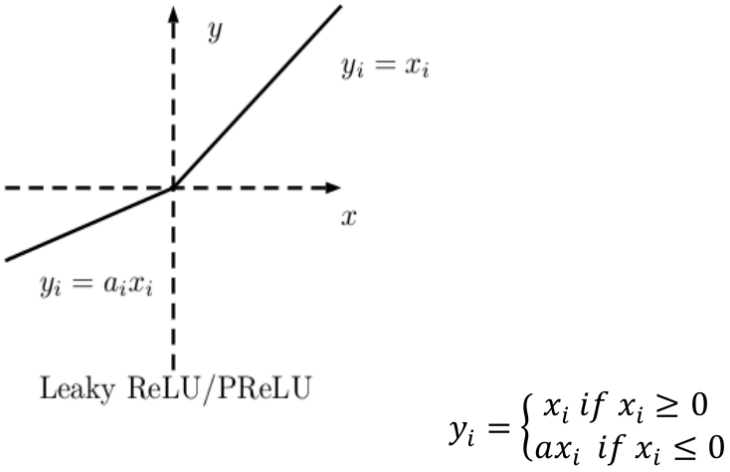


FIGURE 3.4 – *fonction d'activation Leaky ReLu*

classes (Feu, fumée et Arrière-plans) *Équation [103]* . Cette fonction permet de normaliser les valeurs de sorties fournissant une probabilité d'appartenance à l'une des trois classes.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{3} e^{z_i}} \tag{3.1}$$

3.3 Entraînement

L'ajustement des poids du réseau est réalisé par une descente stochastique de gradient par mini-lots brassés (mini-batches) de 100 images. Le gradient est calculé pour totalité des images de chaque mini-lot *algorithme 3.1*. L'objectif du brassage des données d'apprentissage est d'éviter de rester coincé dans un minimum local.

ALGORITHM 3.1 – Entrainement du réseau par une descente de gradient stochastique

- Initialiser chaque paramètre du réseau Θ du réseau avec de faibles valeurs aléatoires. $k = 1$ Itération.
- définir les règles d'évolution du taux d'apprentissage η
- **While** tant le critère fixé n'est pas atteint **faire**
  - mélanger les données.
  - Pour les m images des N mini-lots de la base d'apprentissage $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$ et des données attendues $\{y^{(1)}, y^{(2)}, ..., y^{(m)}\}$
    * estimer le gradient de l'erreur E : $\mathbf{g} = \frac{1}{m} \nabla_\Theta \sum_i E(f(x^{(i)}, \Theta); y^{(i)})$
    * mise à jour des paramètres du réseau $\Theta_n = \Theta_{n-1} - \eta_k \mathbf{g}$
  - k=k+1 itération suivante.

  fin while

L'utilisation des mini-lots permet de travailler avec un gradient estimé sur plusieurs exemples réduisant le bruitage lors de la phase d'apprentissage. Si la taille des mini-lots est trop importante, on tend vers une descente de gradient classique. Enfin, l'utilisation de mini-lot réduit le coût de calcul de par la diminution de la mise à jour des paramètres du réseau. Les poids du réseau sont initialisés aléatoirement avec de faibles valeurs. Le

taux d'apprentissage est fixé à 0.01 et décroît toutes les 5 itérations d'un facteur 0.95. Le momentum quant à lui est fixé initialement à 0.9 et augmente toutes les 5 itérations pour atteindre l'unité. La décroissance du taux d'apprentissage permet au début de l'apprentissage de ne pas s'enfermer dans un minimum local. En fin d'apprentissage, le taux diminue pour affiner le modèle. Le coefficient de momentum est un hyperparamètre qui représente l'inertie d'apprentissage. Il permet de limiter l'oscillation en modulant la mise à jour des paramètres du réseau.

L'ensemble des données de validation a pour objectif de suivre le comportement lors de l'entraînement du réseau et éviter le surapprentisage. Toutes les 5 itérations, l'erreur sur les données de validation est calculée (images non utilisées pour l'ajustement des paramètres du réseau). Ces données n'ont pas servi à la mise à jour des paramètres du réseau ce qui donne une bonne indication sur la généralisation de classification du réseau. Si l'erreur d'apprentissage diminue et que l'erreur sur la base de validation augmente, cela indique une spécialisation du réseau sur les données d'apprentissage *Figure 3.5*. Le réseau perd petit à petit sa force de généralisation sur des données extérieures à la base d'apprentissage. Afin de diminuer le surapprentissage, un dropout égal à 0.5 a été introduit dans la phase d'apprentissage [104]. Cette technique éteint aléatoirement un pourcentage des cellules du réseau afin d'éviter une symétrisation du réseau et permet de générer un modèle de données différent pendant la phase d'apprentissage.

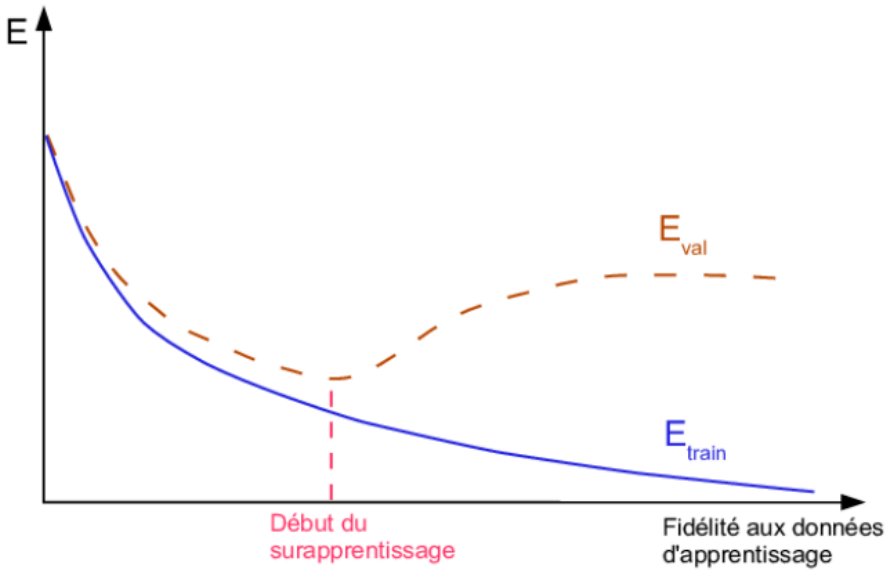


FIGURE 3.5 – *Surapprentissage du réseau*

Des essais successifs ont permit de déterminer les valeurs optimales des hyperparamètres : la taille des mini-lots, le taux d'apprentissage et le coefficient de momentum. L'implémentation de l'algorithme a été réalisée sous Python avec la librairie Theano/Lasagne.

3.4 Critères d'évaluation

Le critère de rappel 3.2 a été choisi et appliqué sur les images de la base de test dans le but d'évaluer la performance de classification du réseau. Il correspond à la proportion de résultats positifs réels identifiés correctement. Nous aurions pu ajouter également les critères d'évaluation connexes que sont la justesse 3.4 ou la précision 3.3.

$$Rappel = \frac{TP}{TP + FN} \tag{3.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

$$Justesse = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.4}$$

Tous les critères retenus se calculent à partir d'une matrice de confusion *Table 3.2*.

TABLE 3.2 – *Matrice de confusion*

		Ground Truth		
		Positive	Negative	***Total***
Predicted	Positive	**TP**	**FP**	*Positives*
	Negative	**FN**	**TN**	*Negatives*
	Total	*GTPos*	*GTNeg*	Pixels image

Nous avons décidé de réaliser une matrice de confusion pour les 3 classes : fumées, feu et arrière-plan. Ces matrices de confusion nous ont permis de calculer la précision, justesse et rappel moyen sur les images de la base de données de test.

Afin de déterminer la précision de chaque classe en fonction du seuil de discrimination, nous avons décidé de tracer les courbes ROC (Receiver Operating Characteristics) *Figure 3.6*. Pour des seuils de discrimination variable de 0 à 1 et pour chaque classe, nous traçons la sensibilité (true positive rate) $\frac{TP}{Positifs}$ en fonction de 1- spécificité (false positive rate) avec la spécificité $\frac{FP}{Negatifs}$. L'aire sous la courbe (AUC) ROC donne une information sur la qualité de la classification. Si l'aire tend vers l'unité, ceci indique une bonne classification.

3.5 Résultats

Les matrices de confusions pour les 3 classes montrent une très bonne classification du feu et de la fumée sur les images de la base test *Tables 3.7, 3.9 et 3.8*. Ces matrices mettent en avant peu de faux positifs ou faux négatifs. Les résultats sur le rappel, la

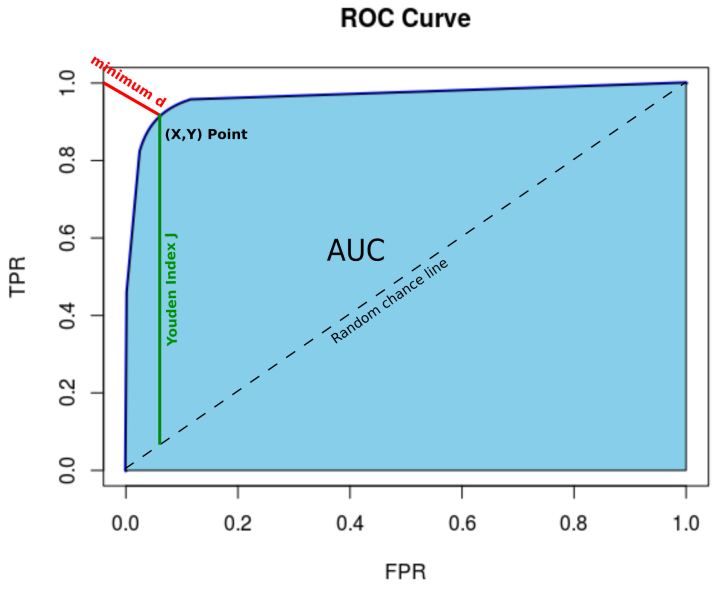


FIGURE 3.6 – *Courbe ROC*

justesse et la précision pour les 3 classes confirment une excellente classification *Table 3.3*. Le rappel moyen sur les 3 classes s'élève à 97,9%. On peut noter une meilleure performance de classification pour la classe feu que pour la classe fumée.

Fire		True class	
		True	*False*
Hypothesis class	**True**	1400	3[a]
	False	27[a]	4154

[a]not smoke images

FIGURE 3.7 – *Matrice de confusion pour le feu - Base de test*

No Fire/Smoke		True class	
		True	*False*
Hypothesis class	**True**	2370	87[a]
	False	29[b]	3098

[a]Image Fire 27 – image Smoke 60
[b]Image Fire 3 – image Smoke 26

FIGURE 3.8 – *Matrice de confusion pour l'arrière-plan - Base de test*

Smoke		True class	
		True	*False*
Hypothesis class	**True**	1698	26[a]
	False	60[a]	3800

[a]not fire images

FIGURE 3.9 – *Matrice de confusion pour la fumée - Base de test*

Classe	rappel (%)	justesse (%)	précision (%)
Feu	98,1	99,5	99,8
Fumée	96,6	98,5	98,5
Background	98,9	97,9	96,4
Moyenne	97,9	98,6	98,2

TABLE 3.3 – *Précision, justesse et rappel sur la base de test*

Les aires sous les courbes ROC pour les 3 classes *Figure 3.6* sont proches de l'unité pointant une excellente classification. La courbe ROC du feu possède une plus grande aire sous la courbe que la fumée et l'arrière-plan. Ceci confirme la valeur importante du critère

de rappel. Cette meilleure classification du feu peut être expliquée par le fait que le feu possède des particularités colorimétriques plus évidentes que la fumée.

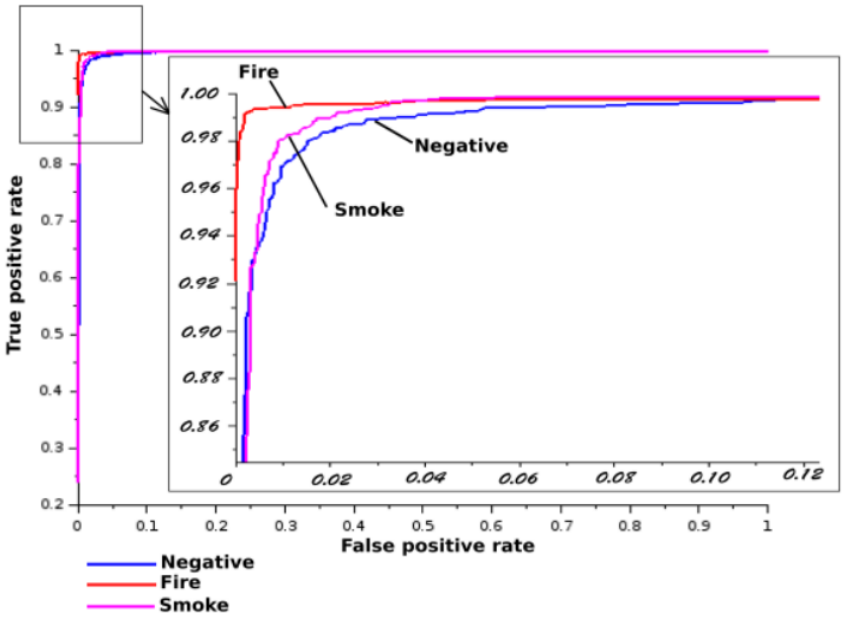


FIGURE 3.10 – *Courbes ROC pour les 3 classes.*

Fort de ces résultats sur des imagettes de 64x64 pixels, nous avons décidé de réaliser une détection et une localisation du feu et de la fumée sur des images complètes. Pour ce faire, nous avons décidé d'utiliser la technique de la fenêtre flottante. L'idée était de faire glisser une fenêtre de 64x64 pixels sur l'image d'entrée en RGB afin de déterminer la classe (feu, fumée ou arrière-plan) de ladite fenêtre. Notre objectif était de réaliser une détection compatible avec du temps réel afin de l'appliquer sur des vidéos. Dans le but d'accélérer le processus de classification de l'image, nous avons découpé notre réseau en deux parties *Figure 3.11*. Une première partie convolutionnelle, composée uniquement d'opération de convolutions et de maxpooling afin d'obtenir une unique carte de caractéristiques. L'obtention de cette dernière carte de caractéristiques n'impose pas de taille de l'image d'entrée. La seconde partie s'articule autour de couches totalement connectées et fixe la taille d'entrée dans celle-ci. Toutefois, l'objectif est de pouvoir analyser des images de tailles diverses et variées.

Or, la fenêtre glissante de 64x64 pixels sur l'image d'entrée est équivalente à une fenêtre de 12x12 pixels sur la dernière couche de caractéristiques de la première partie du réseau *Figure 3.12*. La taille d'entrée dans la seconde partie du réseau était donc de 12x12x1. L'idée fut de transformer la dernière carte de caractéristiques en une matrice de dimensions de taille 12x12x1xN avec N étant le nombre de fenêtres appliquées sur l'image d'entrée. Ce nombre de fenêtres dépend du pas de déplacement de la fenêtre glissante dans l'image d'entrée. Un pas de déplacement de 16 pixels de la fenêtre glissante sur l'image RGB d'entrée correspond à un pas de déplacement de 4 pixels sur la dernière carte de caractéristique de la première partie du réseau.

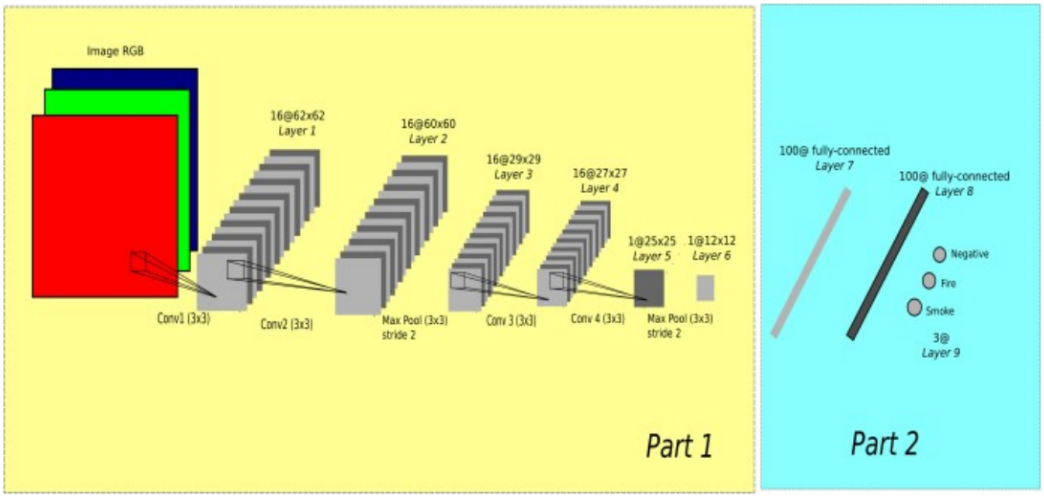


FIGURE 3.11 – *Architecture du réseau en deux parties.*

Il faut noter que le feu est facilement identifiable sur la dernière carte de caractéristiques sans utiliser la seconde partie du réseau *Figure 3.12 image de gauche.*

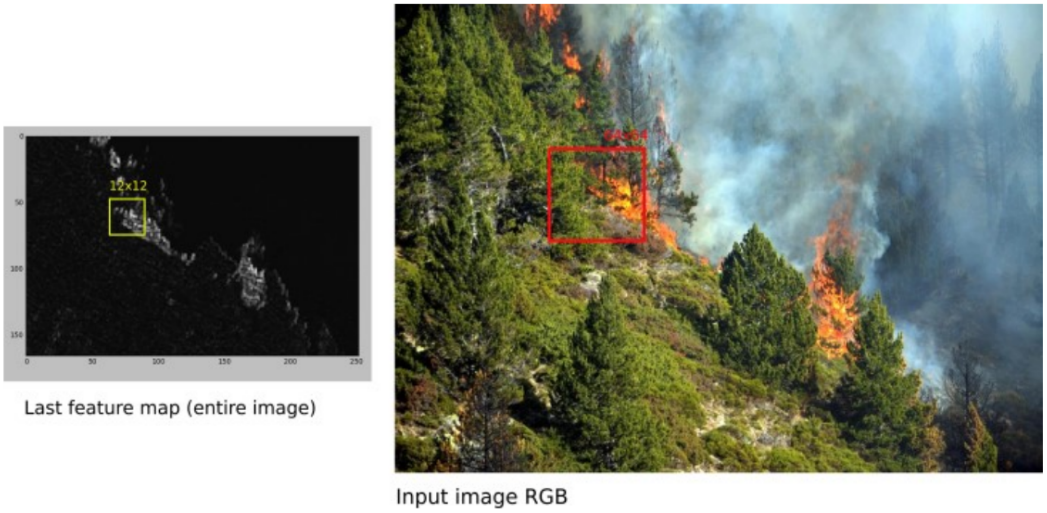


FIGURE 3.12 – *Fenêtre glissante sur l'image d'entrée à droite et sur la dernière carte de caractéristique à gauche.*

Cette technique permet de se libérer de la taille de l'image d'entrée et d'accélérer le processus en utilisant le GPU de la carte graphique de l'ordinateur. La seconde partie du réseau détermine la classe correspondante la plus probable de chaque fenêtre. Il suffira ensuite de reconstruire une carte de probabilité issue de la fonction Softmax de chaque classe et de la superposer à l'image d'entrée en prenant garde à la dimensionner convenablement *Figure 3.13 et 3.14* .

Nous arrivons à une bonne indication des zones de fumées et de feux. Toutefois, la segmentation manque de précision surtout à l'intersection entre deux classes dues à la compression de l'image d'un facteur $\frac{64}{12}$ eme et du pas de déplacement de la fenêtre glissante.

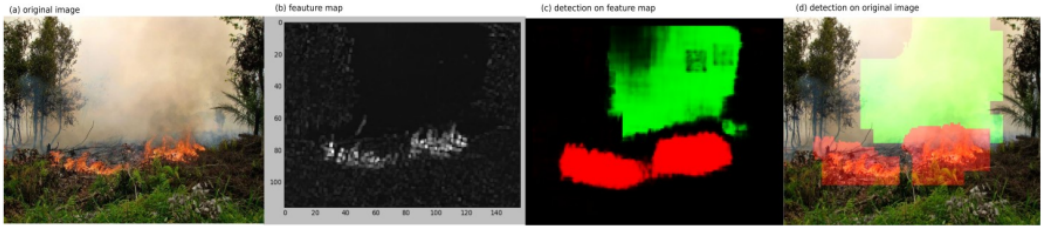


FIGURE 3.13 – *Résultat du glissement de la fenêtre. En vert le masque de fumée et en rouge le masque de feu.*

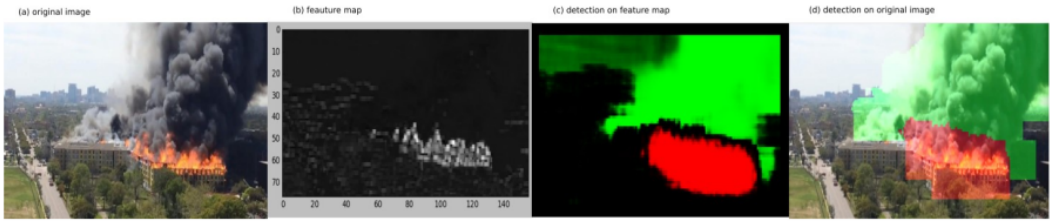


FIGURE 3.14 – *Résultat du glissement de la fenêtre. En vert le masque de fumée et en rouge le masque de feu.*

L'objectif étant la détection et la localisation du feu et de la fumée compatible avec du temps réel, nous avons décidé d'évaluer le temps de prédiction sur des images d'entrées de tailles différentes. Nous avons comparé notre méthode (réalisation d'une unique carte de caractéristiques par la première partie du réseau puis découpage dans celle-ci de fenêtre de 12x12 puis prédiction par la seconde partie du réseau.) à la méthode classique consistant à réaliser une fenêtre glissante sur l'image d'entrée.

Les temps de prédiction sont des temps moyens réalisés sur 200 images de la base de test. Dans un premier temps, nous avons fixé le pas de déplacement de la fenêtre glissante à 16 pixels dans l'image d'entrée correspondant à 4 pixels sur la dernière carte de caractéristiques. Pour différentes tailles d'images d'entrées, nous avons déterminé le temps de prédiction des masques de fumée et de feu avec notre méthode et la méthode consistant à prédire la classe de chaque fenêtre sur l'image d'entrée. Le *Tableau 3.4* montre les temps de prédictions et le ratio entre les deux méthodes. La *Figure 3.15* montre le ratio du temps de prédiction entre les deux méthodes en fonction du nombre de fenêtres prédites. La carte graphique utilisée pour l'expérimentation était une GTX 980 Ti.

Taille image(px)	Nb de fenêtres	Méthode 1 (ms)	Méthode 2 (ms)	ratio
320x240	176	11,1	71,2	6,4
640x480	936	30,1	314	10,4
1280x720	3116	73,1	1000	13,7
1980x1080	7308	159	2327	14,7

TABLE 3.4 – *Temps de prédiction du masque de fumée et de feu. La Méthode 1 correspond à notre méthode (utilisation de la dernière carte de caractéristique). La méthode 2 consiste à prédire la classe de chaque fenêtre d'entrée. Le ratio est le rapport du temps de prédiction entre la méthode 1 et la méthode 2.*

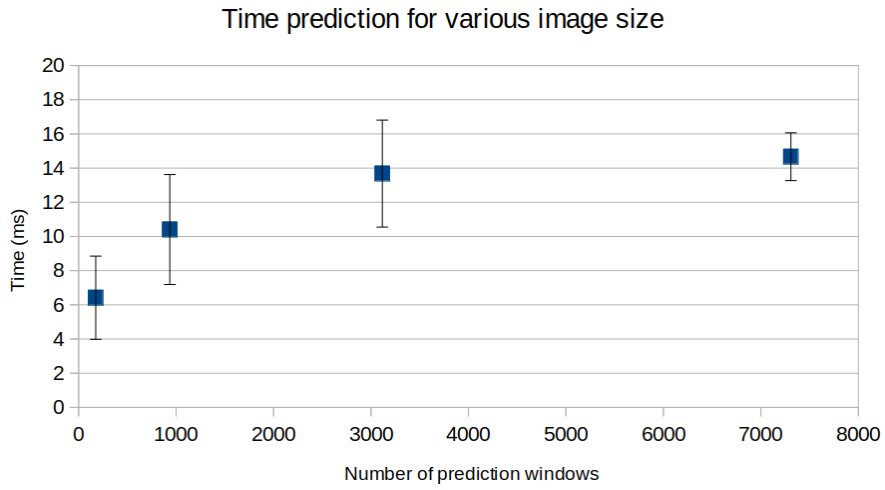


FIGURE 3.15 – *Ratio du temps de prédiction entre la méthode 1 et la méthode 2 en fonction du nombre de fenêtres prédites (taille de l'image d'entrée)*

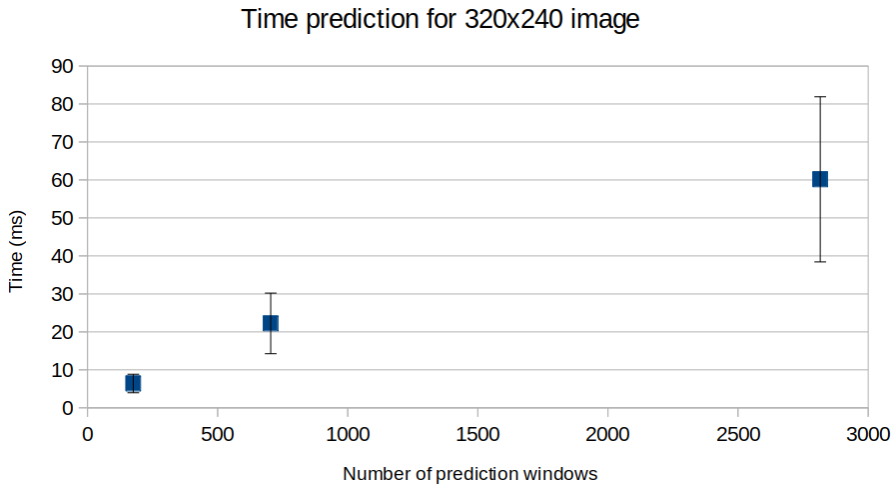


FIGURE 3.16 – *Ratio du temps de prédiction entre la méthode 1 et la méthode 2 en fonction du nombre de fenêtres prédites (taille de l'image d'entrée)*

Dans un second temps, nous avons fixé la taille de l'image d'entrée à 320x240 pixels et appliqué divers pas de déplacement de la fenêtre glissante *Figure 3.16 (Le pas de déplacement de la fenêtre glissante est donné sur l'image d'entrée. Respectivement pour les 3 pas 16,8,4)*

3.6 Conclusion et perspectives

L'utilisation du réseau convolutif pour la classification du feu et de la fumée donne de très bons résultats avec une précision de classification pour la fumée supérieure à 98% et pour le feu supérieur à 99%. La courbe ROC pour le feu et la fumée confirme une très bonne précision de classification. Le feu possède moins de faux positifs et de faux

négatifs que la fumée. Ceci est certainement dû à la caractéristique colorimétrique marquée du feu dans les rouges orangés. Il faut tout de même savoir raison garder quant à cette très bonne performance de classification qui a été testée uniquement sur notre base de données. Pour obtenir une bonne généralisation de classification, il aurait été souhaitable d'augmenter la base de données ou de fusionner diverses bases. Toutefois les performances du réseau utilisées sont excellentes au vu de son architecture légère. Depuis l'utilisation des réseaux convolutifs pour la détection du feu et/ou de la fumée s'est répandue depuis plusieurs années. L'avantage du réseau convolutif réside dans le fait qu'il va, dans sa phase d'entraînement, rechercher sur les trois canaux les relations entre les pixels permettant de classifier une image comme contant du feu et/ou de la fumée. Les méthodes traditionnelles devaient faire appel à un expert du domaine afin de trouver le vecteur caractéristique de l'objet à détecter. Dans le cas du feu, les paramètres étaient principalement liés à la couleur des pixels et à la variation de l'intensité des pixels dans le temps. L'utilisation du réseau convolutif peut se libèrer de l'aspect temporel et permet la détection du feu et/ou de la fumée même lorsque la caméra n'est pas fixe.

L'architecture de notre réseau, avec une la constitution d'une unique carte de caractéristiques avant les couches totalement connectées, permet de réaliser une segmentation en une seule passe du feu et de la fumée sur des images. Pour des images de faible définition et avec la carte graphique GTX 980 TI sous python, la segmentation peut se réaliser dans des temps compatibles avec le temps réel. Pour une image 1280x720, la segmentation du feu et de la fumée peut être réalisée sur une vidéo à 10 images par seconde.

Toutefois, la segmentation par cette méthode met en évidence des défauts de précision dus au pas de déplacement de la fenêtre glissante. Des chevauchements s'opèrent entre les prédictions des fenêtres engendrant des faux négatifs surtout entre des zones concomitantes contenant du feu et de la fumée *Figure 3.13 et 3.14*. La segmentation manque de définition à cause de la taille de la fenêtre glissante et du pas de déplacement de celle-ci.

La segmentation est une technique intéressante permettant de localiser et de dimensionner le feu ainsi que la fumée. Le temps réel reste également une priorité dans le cadre d'une surveillance des incendies sur terre ou dans les airs. Il nous a semblé opportun de nous pencher plus particulièrement sur un réseau convolutif profond permettant de segmenter le feu et la fumée de manière plus précise.

Chapitre 4: Segmentation

Contents

La segmentation d'images est une technique consistant à créer des masques binaires contenant les objets recherchés. Elle permet en une seule passe de détecter et de localiser plusieurs objets d'une même classe ou de classe différentes. Segmenter le feu et/ou la fumée de manière précise permet aux pompiers d'avoir une idée de l'importance du sinistre et d'organiser les opérations de secours. La vitesse de segmentation peut fortement varier en fonction de l'architecture du réseau et de la précision de segmentation recherchée. Les réseaux convolutifs permettent de réaliser la segmentation d'objets dans des images avec une grande précision. La base de données reste toutefois un élément très important dans ce type d'apprentissage supervisé.

4.1 Base de données

Comme dans tous les apprentissages, la base de données revêt une importance primordiale. La base doit être suffisamment riche pour une bonne généralisation de la segmentation du feu et de la fumée. Une base d'images de segmentation est lourde à réaliser. La première étape consiste à récupérer des images contenant du feu et de la fumée sur internet. Nous avons veillé à récupérer des images possédant une large plage de niveau de gris et colorimétrique de la fumée. Nous avons privilégié les images aériennes provenant de drones ou d'hélicoptères, car notre objectif est de segmenter des images à l'aide d'une caméra embarquée compatible avec du temps réel. Les 366 images sélectionnées possédaient des tailles différentes. Après avoir dimensionné l'ensemble des images, nous avons découpé celles-ci en deux groupes aléatoirement : 82% soit 300 images pour la base d'entraînement du réseau et 18% soit 66 images pour la base de validation/test. L'objectif de séparer les images de la base d'entraînement et de validation avant la phase d'augmentation d'images est d'être certain que la base de validation n'aura jamais été vue par le réseau. Afin de réaliser les masques, nous avons utilisé le logiciel Labelme [105]. Ce logiciel permet de réaliser manuellement les polygones des objets de chaque classe présents dans l'image. Un utilitaire sous Python utilisant la librairie Opencv à été réalisé pour obtenir les masques des 3 classes (0 pour l'arrière-plan, 1 pour la fumée et 2 pour le feu). Pour chaque image, nous avons réalisé manuellement les polygones contenant les feux et la fumée. La difficulté a été parfois de déterminer les limites du feu et de la fumée sur les images. Lorsque le feu se voyait par transparence sous la fumée, fallait-il le segmenter comme un feu ou comme de

la fumée. De la même manière lorsque la fumée se distinguait très faiblement devions-nous en tenir compte ou pas. Toutes ces questions ont certainement amené à des erreurs de segmentation de la base de référence (groundtruth).

La seconde étape se résume à augmenter le nombre d'images, car 366 images ne suffisent pas pour ajuster les paramètres d'un réseau de segmentation. Nous avons réalisé un petit utilitaire sous Python avec la librairie Opencv afin de réaliser des opérations telles que des miroirs, des rotations, des zooms, des modifications de la luminosité, du contraste et l'ajout de bruit gaussien ou une combinaison de ces transformations sur toutes les images ainsi que les masques correspondants. Nous avons ainsi pu atteindre 7224 images pour la base d'entraînement du réseau et 1560 pour la base de validation.

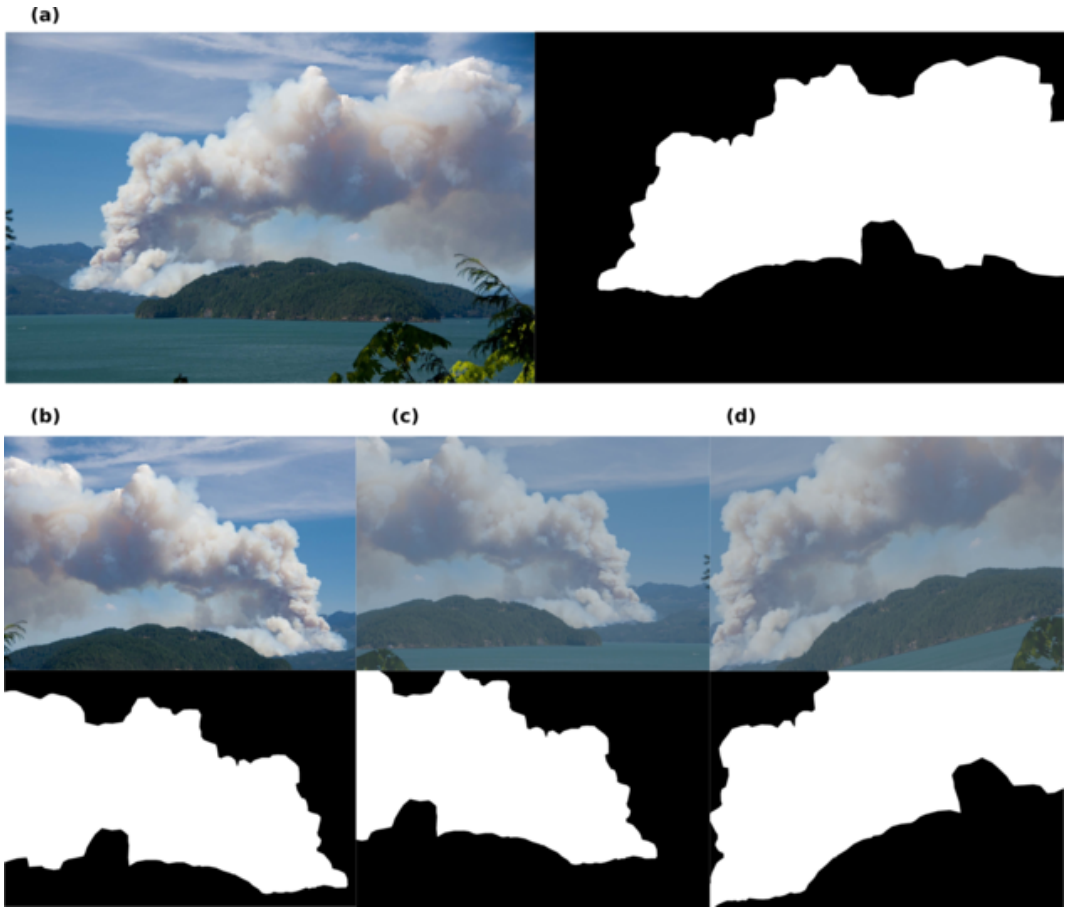


FIGURE 4.1 – *Exemples d'augmentation d'images pour la base de données. a - image et masque de la fumée originels. b - zoom. c - Effet miroir et variation du contraste. d - Rotation et variation du contraste.*

4.2 Architecture du réseau

Le réseau de segmentation devait pouvoir segmenter des images de tailles non fixées, compatible avec du temps réel, possédant une bonne précision. Nous nous sommes penchés sur les architectures rapides comme YOLO [106], mais l'encadrement du feu avec une boite manquait de précision au vu de la forme non fixe du feu. La structure du réseau U-Net [86]

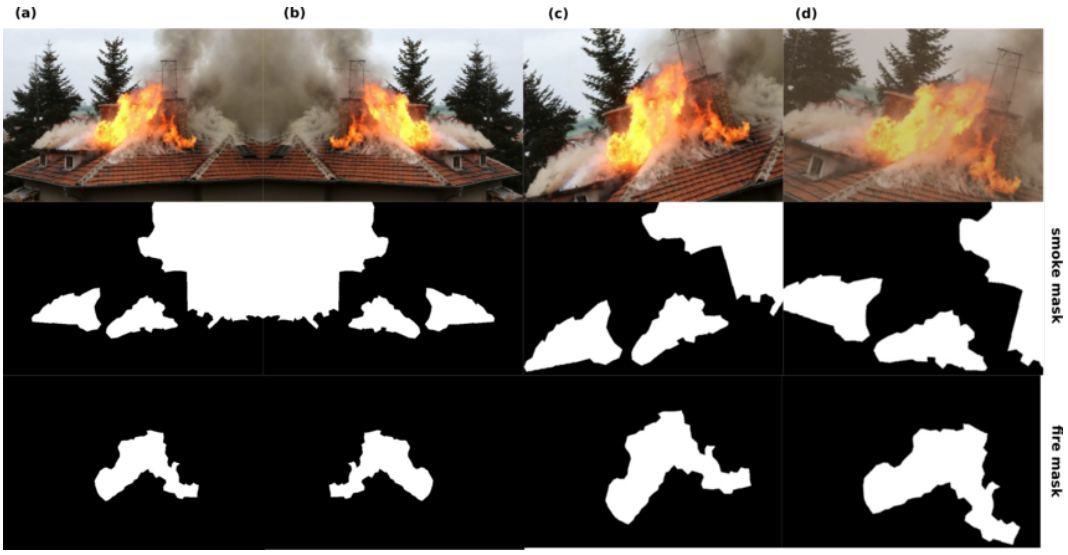


FIGURE 4.2 – *Exemples d'augmentation d'images pour la base de données. a - image et masque de la fumée et du feu originels. b - effets miroir. c - effet de rotation. d - effet de rotation et modification du contraste.*

quant à elle proférait une très bonne précision de segmentation de par l'idée de fusionner les cartes de caractéristiques du chemin de codage avec celui de décodage. Malheureusement, le temps de calcul n'était pas compatible avec du temps réel, même sur des images de basse résolution. Nous avons décidé de trouver une architecture hybride répondant à nos exigences. Afin de gagner en temps d'entraînement du réseau et de limiter le nombre d'images de la base de données, nous avons décidé de baser le chemin de codage sur une architecture VGG16 [74], architecture largement utilisée et possédant des poids pré-entraînés sur la base de ImageNet [75]. Avant d'obtenir la structure finale, de nombreux réseaux ont été testés contenant des blocs résiduels, de multiples tailles de noyaux de convolutions avec des opérations de convolution en parallèles à l'image des blocs d'inception. Nous avons cherché à augmenter le champ réceptif en utilisant des convolutions à trou tant pour le chemin de codage que de décodage. Le réseau le plus performant sur la base de validation était le plus simple, pas de blocs résiduels, d'inception et possédait principalement des opérations de convolution avec des noyaux de dimension 3x3.

Le réseau ainsi sélectionné est constitué de deux chemins *Figure 4.3*. Un chemin de codage de l'information basé sur le réseau VGG16 et constitué d'opérations successives de convolution avec un noyau de 3x3 et de max-pooling. Le codage se termine par une dernière opération de convolution avec un noyau de 7x7. Le chemin de décodage se compose d'opérations de déconvolutions (transpose convolution) correspondant à la fonction inverse des opérations de convolutions. Elles permettent d'augmenter la taille des images d'un facteur déterminé. Les cartes de caractéristiques obtenues par ces opérations de déconvolutions sont ajoutées à des copies des cartes de caractéristiques du chemin de codage. Cette opération permet de propager les informations contextuelles vers les couches supérieures du réseau et ainsi de ne pas perdre d'information importante lors de la reconstruction des masques [107]. Le détail des couches du réseau est donné dans les *tableaux 4.1 et 4.2* (la

taille de l'image d'entrée et des diverses cartes de caractéristiques sont définis pour une image d'entrée de taille 648x480 pixels en RGB). Les fonctions d'activation choisies pour les chemins de codage et de décodages sont des Rectified Linear Unit (ReLU). La dernière carte de caractéristiques du chemin de décodage subit une fonction Softmax *Equation 3.1* permettant de déterminer pour chaque pixel sa classe d'appartenance la plus probable.

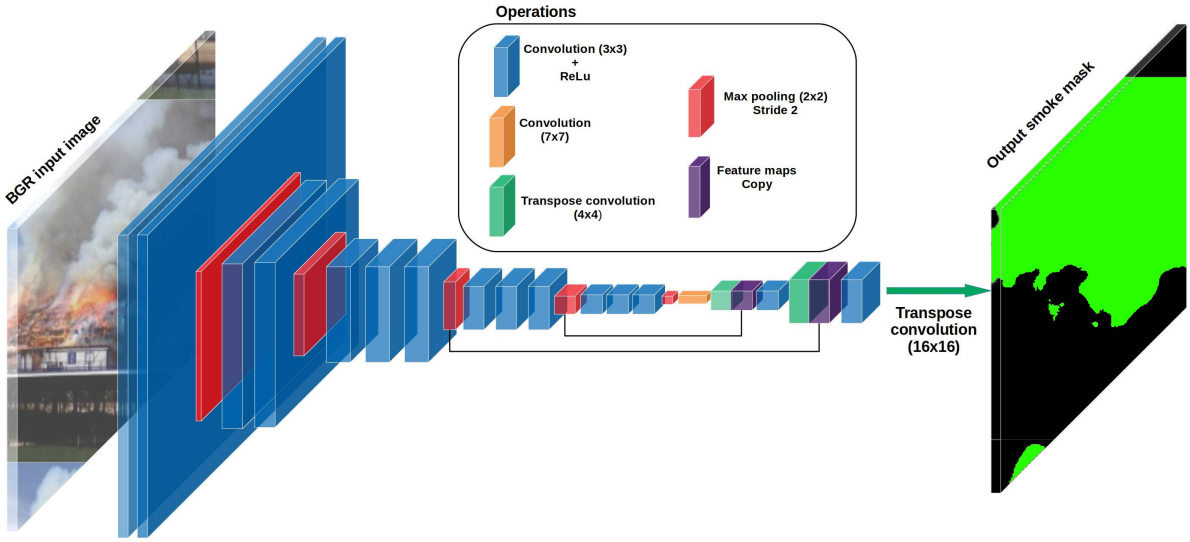


FIGURE 4.3 – *Architecture du réseau sélectionné pour la segmentation*

Afin de pouvoir nous libérer de la taille d'entrée des images dans le réseau, celui-ci ne possède pas de couches totalement connectées. Nous avons pris garde à ce que le temps de création des masques de feu et de fumée soit compatibles avec du temps réel pour des définitions d'images moyennes (640x480).

Nous avons décidé de comparer les performances de notre réseau avec des réseaux existants dans la littérature. Nous avons choisi le réseau U-Net [86] et le réseau créé par Yuan et son équipe spécialisée dans la détection de la fumée [108].

Yuan et al. ont proposé une architecture réseau dédiée à la segmentation de la fumée *Figure 4.4*. Il est composé de deux chemins en parallèle aboutissant à une fusion permettant de créer le masque de la fumée. Les deux chemins sont basés sur une structure codeur-décodeur. Les parties codeur sont organisées autour d'une structure VGG16 et une partie des cartes de caractéristiques du codeur sont injectées dans le décodeur. Les deux chemins possèdent des profondeurs différentes, le premier peu profond permet de conserver des détails de localisation de l'information souvent perdus dans des réseaux profonds. Le second plus profond permet de conserver l'information globale permettant une bonne généralisation [109]. Initialement, le réseau a été entraîné sur des images synthétiques (images de la fumée créées synthétiquement et fusionnées avec des images d'arrière-plan réelles).

Chemin de codage

Réseau de segmentation basé sur VGG16

Operations	*Nb FM*	*name*	*Size FM*
Convolution + ReLu 3x3	64	FM1	640x480
Convolution 3x3 + ReLu	64	FM2	640x480
MaxPooling 2x2	64	FM3	320x240
Convolution 3x3 + ReLu	128	FM4	320x240
Convolution 3x3 + ReLu	128	FM5	320x240
MaxPooling 2x2	128	FM6	160x120
Convolution 3x3 + ReLu	256	FM7	160x120
Convolution 3x3 + ReLu	256	FM8	160x120
Convolution 3x3 + ReLu	256	FM9	160x120
MaxPooling 2x2	256	FM10	80x60
Convolution 3x3 + ReLu	512	FM11	80x60
Convolution 3x3 + ReLu	512	FM12	80x60
Convolution 3x3 + ReLu	512	FM13	80x60
MaxPooling	512	FM14	40x30
Convolution 3x3 + ReLu	512	FM15	40x30
Convolution 3x3 + ReLu	512	FM16	40x30
Convolution 3x3 + ReLu	512	FM17	40x30
MaxPooling 2x2	512	FM18	20x15
Convolution 7x7	1024	sortie chemin de codage	20x15

TABLE 4.1 – *Composition du chemin de codage Nb FM : nombre de cartes de caractéristiques. Size FM : taille de la carte de caractéristiques. Exemple pour une taille de l'image d'entrée du réseau de 640x480 RGB*

Chemin de décodage

Operation type	*Nb FM*	*Size FM*
sortie du chemin de codage	1024	20x15
Deconvolution 4x4 Kernel	512	40x30
Concatenation with FM14	1024	40x30
Convolution 3x3 kernel + ReLU	512	40x30
Deconvolution 4x4 Kernel	256	80x60
Concatenation with FM10	512	80x60
Convolution 3x3 kernel + ReLU	256	80x60
Deconvolution 16x16 Kernel	3	640x480

TABLE 4.2 – *Notre réseau de décodage.Nb FM : nombre de cartes de caractéristiques. Size FM : taille de la carte de caractéristiques. Exemple pour une taille de l'image d'entrée du réseau de 640x480 RGB*

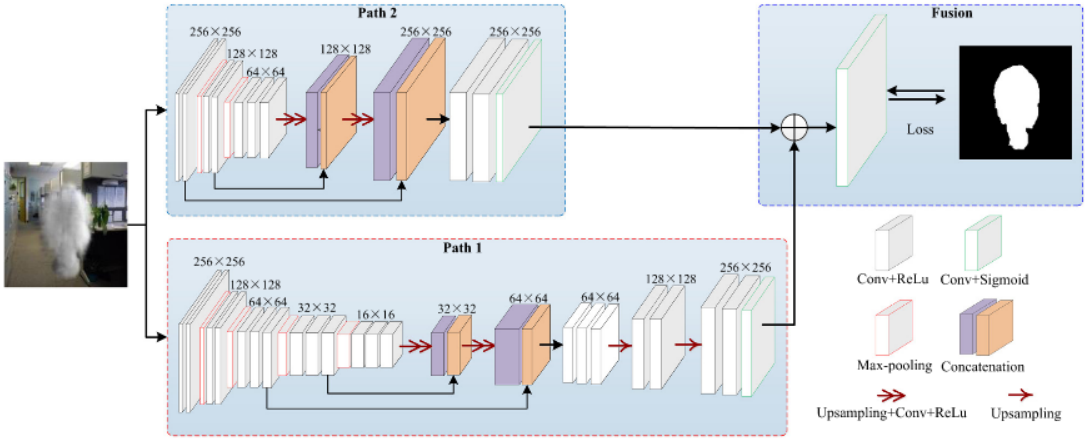


FIGURE 4.4 – *Architecture du réseau de Yuan*

4.3 Entraînement

Notre réseau a été programmé en Python avec la librairie Tensorflow 1.12.0 et Opencv 3.4.0 sous un environnement Linux Ubuntu 18.04. Les phases d'entraînement et de validation ont été réalisées sur le GPU d'une carte graphique Nvidia GeForce 1080. Nous utilisons les poids d'un modèle pré-entraîné sur la base de données Imagenet afin d'accélérer la séquence d'apprentissage.

L'apprentissage se fait sur le jeu de la base d'apprentissage à l'aide de l'algorithme d'optimisateur Adam [101]. Le pas d'apprentissage est fixé à 5.10^{-5} et les données mélangées sont présentées par deux avant de réaliser une rectification des poids du réseau.

L'application Tensorboard de Tensorflow permet de suivre l'évolution de la phase d'entraînement afin d'éviter le sur-aprentissage des paramètres du réseau. Le nombre d'itérations a été fixé empiriquement.

4.4 Critères d'évaluation

Les critères [110] utilisés pour évaluer la qualité de la segmentation du feu et de la fumée sont au nombre de quatre et sont basés sur la matrice de confusion *Table 4.3*. Nous n'avons pas choisi de calculer pour les critères d'évaluation une valeur moyenne pour toutes les classes, mais une valeur moyenne sur tous les échantillons par classe car notre base n'était pas équilibrée. La base de données possède moins de pixels de feu que de fumée. Ceci représente bien la réalité, car le premier indice d'un incendie est le plus souvent la fumée. Le feu vient ensuite, il existe donc toujours une disproportion entre la surface dédiée au feu et à la fumée dans une image d'incendie.

On note TP_i, TN_i, FP_i and FN_i respectivement les nombres de vrais positifs, de vrais négatifs, de faux positifs et de faux négatifs de la i^{eme} image de la base d'entraînement.

- **La justesse** : La justesse est un bon outil permettant de reporter les pixels convenablement classés dans leur classe C.

TABLE 4.3 – *Matrice de confusion*

		Ground Truth: Positive	Ground Truth: Negative	***Total***
Predicted	Positive	**TP**	**FP**	*Positives*
	Negative	**FN**	**TN**	*Negatives*
	Total	$GTPos$	$GTNeg$	Pixels image

$$\overline{Justesse}^c = \sum_{i=1}^{N} \left(\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \right) \tag{4.1}$$

- **La précision** : La précision fournit un évaluation des pixels correctement classifiés par rapport aux prédictions.

$$\overline{Precision}^c = \sum_{i=1}^{N} \left(\frac{TP_i}{TP_i + FP_i} \right) = \sum_{i=1}^{N} \left(\frac{TP_i}{PredPositives_i} \right) \tag{4.2}$$

- **Le rappel** : Le rappel quand à lui fournit une évaluation des pixels bien classés par rapport à la vérité de terrain.

$$\overline{Recall}^c = \sum_{i=1}^{N} \left(\frac{TP_i}{TP_i + FN_i} \right) = \sum_{i=1}^{N} \left(\frac{TP_i}{Truepositives_i} \right) \tag{4.3}$$

Nous avons choisi d'utiliser également le critère IoU (Intersection over Union) prénommé aussi index de Jaccard qui reste le critère d'évaluation le plus populaire dans les problèmes de segmentation. Pour une segmentation parfaite, ce critère tend vers l'unité. De faibles valeurs indiquent une mauvaise segmentation. Ce critère est un critère regroupant la précision et le rappel *Figure 4.5* .

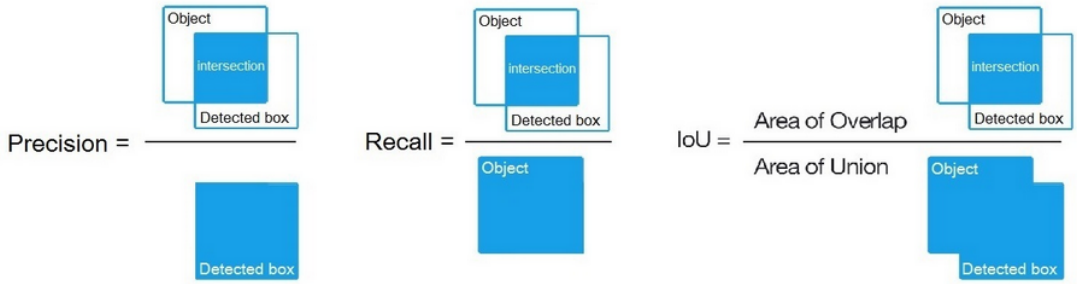


FIGURE 4.5 – *Schéma explicatif des critères IoU, précision et Recall*

Nous avons décidé de tracer les courbes de ROC (Receive Operation Characteristic)

[111] qui définissent la performance de la classification en fonction du seuil de détermination de la classe du pixel *Figure 4.6*. La fonction Softmax en sortie de réseau définit des probabilités d'appartenance à la classe feu, fumée ou arrière-plan. Plus cette probabilité est forte et plus ce pixel a des chances d'appartenir à la classe considérée. La courbe trace pour différents seuils le taux de vrais positifs (proportion des pixels convenablement prédits appartenant à la classe C par rapport aux pixels appartenant réellement à la classe c (vérité de terrain)) en fonction du taux de faux positifs (proportion de pixels mal prédits appartenant à tort à la classe c sur la totalité des pixels n'appartenant pas réellement à la classe c (vérité de terrain)). Si l'aire sous la courbe ROC se rapproche de l'unité, cela indique une très bonne classification des pixels [112].

$$TPR = \sum_{Validation} \sum_{pixels} \frac{TP}{TP + FN} = \sum_{Validation} \sum_{pixels} \frac{TP}{GTPos}$$

$$FPR = \sum_{Validation} \sum_{pixels} \frac{FP}{FP + TN} = \sum_{Validation} \sum_{pixels} \frac{FP}{GTNeg}$$

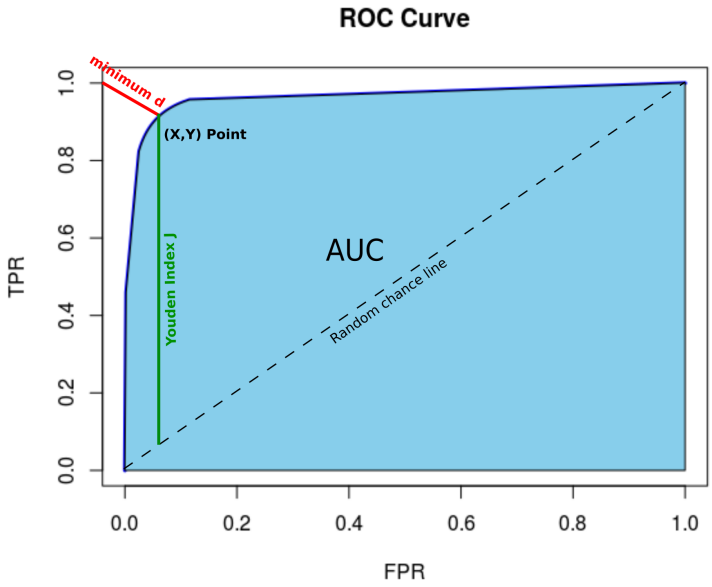


FIGURE 4.6 – *Courbe ROC - youden index J et d distance*

L'inconvénient du critère d'aire sous la courbe ROC est qu'il ne permet pas de déterminer le seuil optimum qui donne le maximum de pixels convenablement prédits. Deux critères assez similaires nommés la distance d et l'index de Youden J donnent une information sur le seuil optimum de prédiction d'une classe.

- **La distance d** : Elle définit la distance entre la courbe et le point (FPR=0 et TPR=1) et un point sur la courbe ROC *Figure 4.6 segment en rouge*. Le point (FPR=0 et TPR=1) correspond à une classification parfaite, uniquement des vrais positifs, sans faux positif. Plus d est faible et meilleure sera la classification des pixels et donc la segmentation du feu et de la fumée. Le seuil de classification possédant

la plus petite distance correspond au seuil générant la meilleure segmentation.

- **L'index de Younden J** [113] : Il définit la distance verticale entre la ligne représentant un système de classification aléatoire *(droite en pointillé sur la Figure 4.6)* des pixels en moyenne et la courbe ROC *(segment en vert sur la Figure 4.6)*. Une valeur importante de cette distance augure d'une bonne classification des pixels. Le seuil de classification possédant la plus grande index J correspond au seuil générant la meilleure segmentation. Ce critère est souvent utilisé, car J maximum permet de maximiser les vrais positifs et minimiser les faux positifs [114].

Enfin, tracer l'IoU ou la justesse en fonction du seuil de prédiction nous a semblé être un indicateur permettant de déterminer la qualité de la classification des pixels. L'allure de la courbe apporte des informations sur la robustesse de la classification des pixels.

4.5 Résultats

Toutes les données de cette section ont été obtenues à partir des données de validation de notre base de données. Il est à noter que les données de validation n'ont jamais servis à ajuster les paramètres du réseau pendant la phase d'entraînement. Nous avons souhaité réaliser une comparaison des performances de la segmentation entre les réseaux de Yuan, U-net et notre réseau. Les trois réseaux ont été entraînés sur notre base de données.

Arrière plan				
	justesse	précision	rappel	IoU
Notre réseau	**0.934**	**0.916**	**0.939**	**0.864**
Réseau U-Net	0.902	0.872	0.915	0.806
Réseau Yuan	0.924	0.899	0.934	0.846

TABLE 4.4 – *Justesse, précision, rappel et IoU pour la classe arrière plan*

Fumée				
	Justesse	précision	rappel	IoU
Notre réseau	**0.925**	**0.941**	**0.907**	**0.858**
Réseau U-Net	0.893	0.915	0.866	0.801
Réseau Yuan	0.916	0.934	0.895	0.841

TABLE 4.5 – *Justesse, précision, rappel et IoU pour la classe fumée*

Les *Tables 4.5, 4.6 et 4.4* montrent pour tous les réseaux une moins bonne qualité de segmentation pour le feu que pour la fumée et l'arrière-plan. Ceci est surprenant, car le feu possède des caractéristiques colorimétriques plus marquées que la fumée. Dans les images

Feu				
	Justesse	précision	rappel	IoU
Notre réseau	**0.981**	0.794	**0.890**	**0.723**
Réseau U-Net	0.977	0.764	0.833	0.663
Réseau Yuan	**0.981**	**0.813**	0.860	0.718

TABLE 4.6 – *Justesse, précision, rappel et IoU pour la classe feux*

de la base de données, le feu possède une teinte tirant vers le rouge orange. La première explication provient de la création manuelle des masques. Il est parfois difficile de discerner le feu de la fumée. Dans le cas de présence de feu derrière la fumée, devons-nous le classer en feu ou en fumée. Sur la *Figure 4.7*, la personne qui a segmenté le feu a privilégié la fumée par rapport au feu et n'a segmenté le feu uniquement s'il n'y avait pas de fumée.

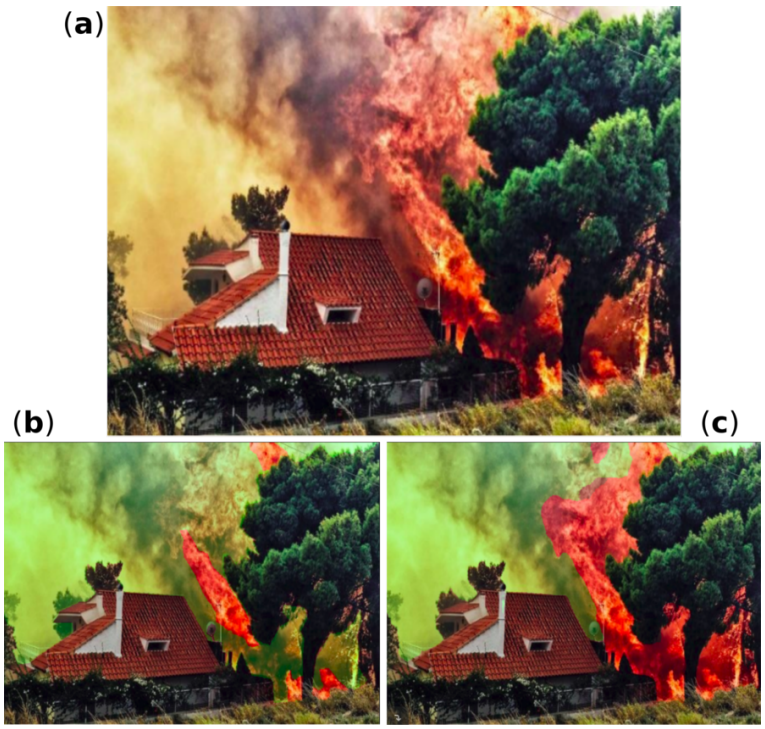


FIGURE 4.7 – *Erreur de segmentation du feu. (a) image RGB - (B) segmentation manuelle du feu (groundtruth) - (C) prédiction de la segmentation du feu en rouge*

La seconde pourrait provenir du déséquilibre de pixels considérés comme appartenant au feu par rapport à ceux de la fumée. Afin de déterminer si cette plus faible faculté de classifier le feu provient de ce déséquilibre, nous avons décidé d'entraîner le réseau en prenant en compte le poids de chaque classe dans les pixels de la base d'entraînement [115]. Pour ce faire, nous avons déterminé le poids relatif des pixels de chaque classe et utilisé une fonction perte cross entropie modulée par le poids de chaque classe *Table 4.7*.

$$w_c = \frac{median f_c}{freq_c}$$

Avec $freq_c$ le nombre total de pixels de la classe c divisé par le nombre total de pixels dans toutes les images de la base d'entraînement. $median f_c$ correspond à la médiane de cette fréquence pour la classe c.

	Arrière plan	**Fumée**	**Feu**
$median\_fc$	0.510	0.418	0.064
f_c	0.503	0.435	0.121
w_c	1.01	0.962	0.529

TABLE 4.7 – *Poids des pixels des 3 classes*

L'objectif est de pénaliser plus fortement les classes majoritaires, ici les pixels de fumée et d'arrière-plan, par rapport à la classe minoritaire qui dans notre cas sont les pixels du feu. Cette technique diminue le sur-aprentissage de la classe dominante. L'entraînement de notre réseau avec la modulation du poids de chaque classe n'apporte pas une augmentation nette de la performance de classification des pixels du feu *Table 4.8*. L'IoU de la segmentation du feu subit une légère augmentation, pendant que les deux autres classes baissent légèrement.

	Justesse	Précision	Rappel	IoU
Arrière plan				
Notre réseau	**0.934**	**0.916**	**0.939**	**0.864**
Notre réseau pondéré	0.931	0.908	0.943	0.860
Fumée				
Notre réseau	**0.925**	0.941	**0.907**	**0.858**
Notre réseau pondéré	0.923	**0.942**	0.901	0.854
Feu				
Notre réseau	0.981	0.794	**0.890**	0.723
Notre réseau pondéré	**0.983**	**0.819**	**0.890**	**0.744**

TABLE 4.8 – *Comparaison entre l'entrainement du réseau avec et sans prise en compte des poids de chaque classe (cross-entropy loss) - Average justesse, precision, recall et IoU.*

Les *Tables 4.5, 4.6 et 4.4* mettent en évidence une meilleure qualité de segmentation de la fumée pour notre réseau que pour les réseaux U-Net et Yuan. Toutefois, les résultats du réseau de Yuan sont proches de nos résultats.

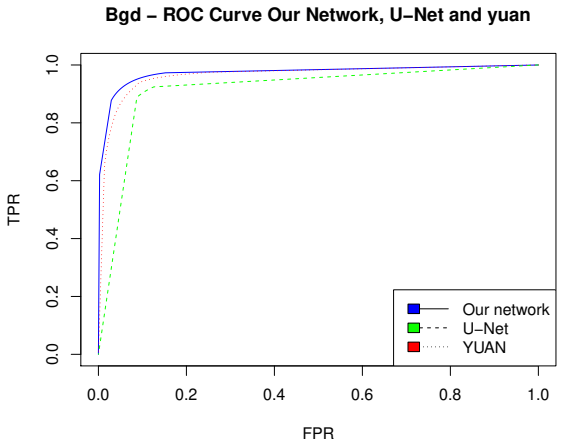


FIGURE 4.8 – *Courbes ROC pour la classe arrière plan*

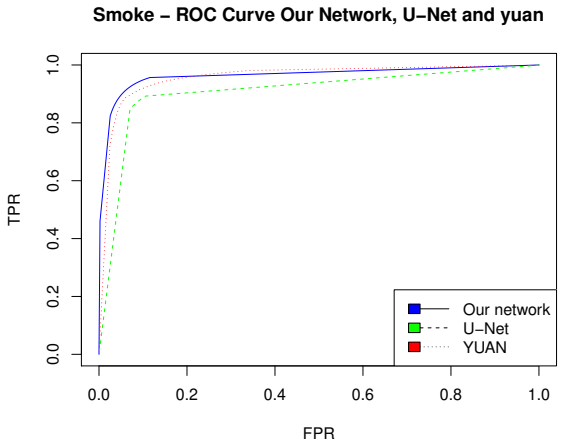


FIGURE 4.9 – *Courbes ROC pour la classe arrière plan*

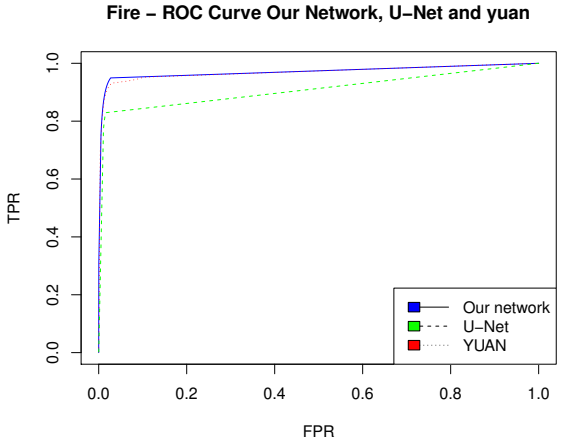


FIGURE 4.10 – *Courbes ROC pour la classe Feux*

Aires sous les courbes ROC (AUC)

	AUC Background	AUC Smoke	AUC Fire
Our network	**0.973**	**0.963**	**0.970**
U-Net network	0.914	0.906	0.908
Yuan network	0.964	0.958	0.969

TABLE 4.9 – *AUC de la courbe ROC pour les classes background, fumée et feux*

Les courbes ROC de notre modèle et celui de Yuan montrent une excellente classification des pixels pour les trois classes. Les aires sous les courbes de la *Table 4.9* des deux réseaux cités précédemment sont proches de l'unité et confirment la bonne qualité de la segmentation sur les 3 classes. Toutefois, notre réseau produit une courbe ROC atteignant

plus rapidement l'unité que celui de Yuan, indiquant un moins grand taux de faux positifs et de facto une meilleure segmentation du feu et de la fumée.

Nous avons souhaité compléter notre étude de la qualité de segmentation par le tracé de la justesse en fonction du seuil de prédiction de classification des pixels pour les classes feu et fumée. L'idée est de déterminer par ce tracé si les probabilités de classification des pixels pour une classe donnée sont importantes.

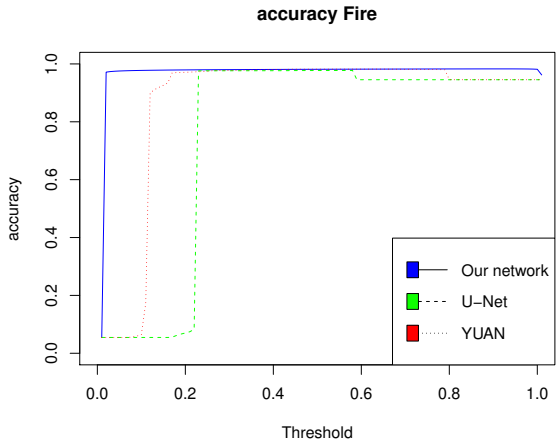


FIGURE 4.11 – *Justesse en fonction du seuil de prédiction pour la classe feu pour notre réseau, U-Net et Yuan*

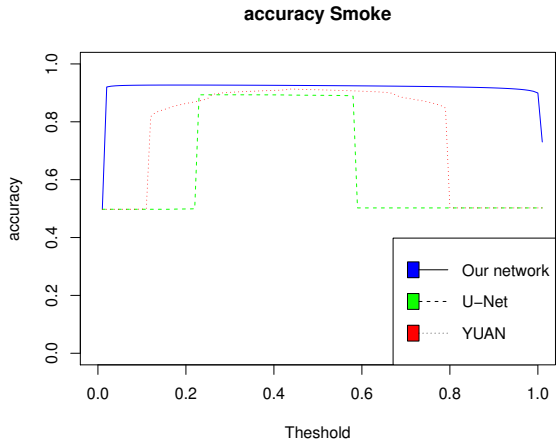


FIGURE 4.12 – *Justesse en fonction du seuil de prédiction pour la classe feu pour notre réseau, U-Net et Yuan*

Nous constatons pour notre réseau et pour la classe fumée une courbe possédant un long plateau à des valeurs proches de l'unité. La courbe retombe pour des seuils proches également de l'unité *Figure 4.12*. Nous pouvons interpréter cette allure par une très forte probabilité de prédiction des pixels de fumée en adéquation avec la vérité de terrain. Les taux de vrais positifs et vrais négatifs sont très importants même pour de fortes valeurs du seuil, ceci présuppose que la grande majorité des pixels prédits de la classe fumée possède une forte probabilité d'appartenance à cette classe. Les réseaux Yuan et U-Net quant à eux montrent une décroissance brutale respectivement pour des seuils à 0,8 et 0,6 indiquant qu'une faible partie des valeurs des probabilités de prédiction des pixels de la fumée sont supérieures respectivement à 80% et 60%. On note une allure de courbe et une analyse identique pour les pixels de la classe feu *Figure 4.11* . Le phénomène de décroissance pour les réseaux Yuan et U-Net est beaucoup moins marqué que pour la classe fumée.

L'index de Houden pour notre réseau montre pour les classes feu et fumée un plateau à une valeur indiquant que toutes les données de la courbe ROC sont regroupées à gauche et proches de l'unité *Figures 4.13 et 4.14*. Ceci met en évidence une faible proportion de faux positifs, quel que soit le seuil de prédiction choisi. Pour les réseaux Yuan et U-Net, les courbes possèdent une allure de créneau, montrant une plus grande proportion des faux positifs que notre réseau pour les faibles seuils de prédiction. Nous avons remarqué une grande similitude de comportement entre l'indice de Houden et la distance d. Nous avons

donc décidé de tracer uniquement l'indice de Houden J en fonction du seuil de prédiction.

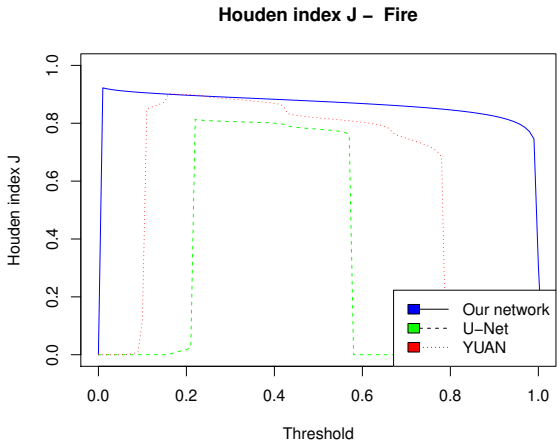


FIGURE 4.13 – *Index de Houden en fonction du seuil de prédiction pour la classe feu pour notre réseau, U-Net et Yuan*

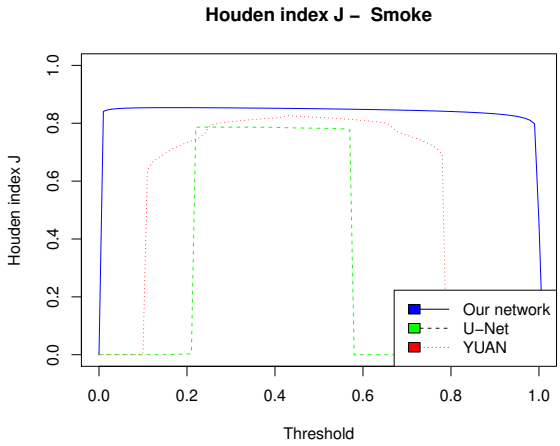


FIGURE 4.14 – *Index de Houden en fonction du seuil de prédiction pour la classe feu pour notre réseau, U-Net et Yuan*

Enfin, nous avons souhaité tracer l'IoU en fonction du seuil de prédiction pour les classes feu et fumée afin de confirmer le comportement de la classification *Figures 4.15 et 4.16*. L'IoU est le critère le plus utilisé pour l'évaluation de la qualité d'une segmentation d'objets. Les courbes de l'IoU pour les classes feu et fumée, pour notre réseau fait apparaitre un large plateau. Ceci confirme les allégations précédentes au sujet d'une forte probabilité de prédiction des pixels feu et fumée. Le large plateau indique une grande proportion de forte probabilité de prédiction pour les deux classes. Nous remarquons une similitude de comportement de la courbe de IoU et de justesse pour la classe fumée. Les réseaux de Yuan et U-Net possèdent des plateaux bien plus réduits avec une décroissance rapide à des seuils respectifs pour de 0,6 et 0,8 indiquant une moindre grande proportion de forte probabilité de prédiction pour les deux classes.

Pour confirmer notre idée de plus grande proportion de forte probabilité de prédiction de notre réseau par rapport aux deux autres réseaux, nous avons sur la totalité des pixels de toutes les images de validation calculé la valeur moyenne et l'écart type de prédiction des vrais positifs pour les classes feu et fumée *Table 4.10*. Les résultats confirment que comparé aux deux autres réseaux, la probabilité de prédiction des pixels feu et fumée pour notre réseau est bien supérieure en moyenne aux résultats des deux autres réseaux. La dispersion autour de cette moyenne par rapport au réseau de Yuan est également meilleure.

Le temps de prédiction des masques revêt une importance capitale dans la détection en temps réel. La *Table 4.11* montre que notre réseau pour des images de moyenne définition (640x480) peut générer les masques de segmentation du feu et de la fumée dans des temps compatibles avec le temps réel avec plus de 20 images par seconde avec notre carte graphique.

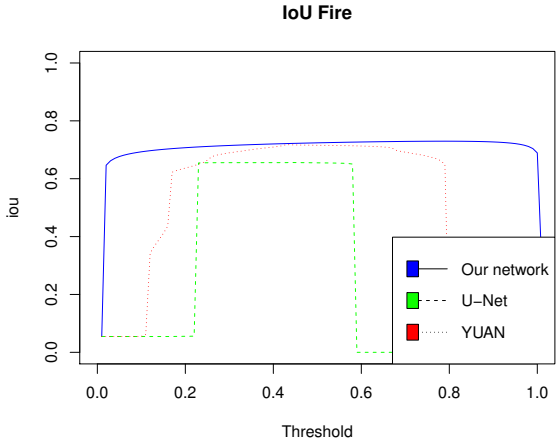


FIGURE 4.15 – *Intersection over Union (IoU) en fonction du seuil de prédiction pour la classe feu pour notre réseau, U-Net et Yuan*

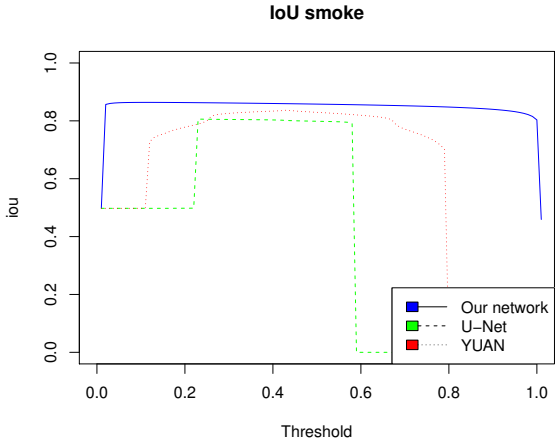


FIGURE 4.16 – *Intersection over Union (IoU) en fonction du seuil de prédiction pour la classe feu pour notre réseau, U-Net et Yuan*

type de réseau	*TP average probabilities*	*TP standard deviation probabilities*
fumée		
Notre réseau	0.987	0.056
U-Net	0.571	0.012
Yuan et al.	0.762	0.068
Feu		
Notre réseau	0.979	0.072
U-Net	0.570	0.020
Yuan et al.	0.757	0.079

TABLE 4.10 – *Probabilité de prédiction et répartition des vraies positifs pous les classes feu et fumée.*

Typoe de réseau	[a] *Nombre de paramètres (millions)*	[b] *Fréquence de segmentation (image/s)*
Our network	57.0	**21.1**
UNet	33.1	11.0
Yuan et al.	29.9	5.8

TABLE 4.11 – [a] *Nombre de paramètres du réseau en million* - [b] *Fréquence de segmentation pour une image 640x480 RGB avec une carte graphique Nvidia GTX1080TI.*

Le fait que notre réseau possède le plus grand nombre de paramètres semble contradictoire avec sa rapidité. La dernière opération de convolution avec un noyau de taille 7x7 du chemin de codage possède la moitié des paramètres du réseau avec 25 millions. Toutefois, cette opération est réalisée sur des cartes de caractéristiques de faible dimension (1/32ème

de la taille de l'image d'entrée) engendrant peu d'opérations de convolutions et donc un temps de calcul faible. De plus, notre réseau possède uniquement 3 opérations de déconvolution contrairement aux réseaux de Yuan et U-Net qui en possèdent respectivement 6 et 4. Notre réseau réalise simplement 2 opérations de convolution pendant la phase de décodage sur des images de faible dimension, contrairement aux réseaux de Yuan et U-Net qui en réalisent 8 dont certaines sur des images proches de la taille des données d'entrées.

La taille du champ réceptif est un paramètre important dans la qualité d'apprentissage d'un réseau convolutif [116]. Un champ réceptif de taille importante permet d'assurer que des informations primordiales de l'image ne sont pas oubliées. Elle permet également de créer une immunité de détection à la translation des objets. Notre réseau de par sa dernière opération de convolution 7x7 possède le plus grand champ réceptif des 3 réseaux étudiés (champ réceptif de 404 pixels pour notre réseau et 140 et 196 respectivement pour les réseaux U-Net et Yuan). Cette taille du champ réceptif important combiné avec la grande profondeur de notre réseau peut expliquer en partie sa bonne segmentation du feu et de la fumée.

La *Figure 2.36* montre des exemples de masques de segmentations pour les 3 réseaux. On peut constater que notre réseau prédit des masques de segmentation pour le feu et la fumée de meilleure qualité que les deux autres réseaux. Le nombre de faux positifs de la fumée est faible même en présence de nuages ou de brume.

4.6 Conclusion et perspectives

Nous avons montré dans ce chapitre la possibilité de segmenter simultanément le feu et la fumée à l'aide d'un réseau convolutif. Le réseau proposé, sur des images de basse résolution, permet la segmentation en temps réel du feu et de la fumée. La profondeur du réseau et la largeur du champ réceptif semblent être des paramètres importants pour obtenir une segmentation de qualité.

La base de données d'entraînement reste un paramètre primordial pour une bonne généralisation. Toutefois, le feu et la fumée sont des objets difficiles à segmenter. Les limites de la fumée et du feu sur une image sont subjectives et donnent lieu à des erreurs de segmentation. Les personnes qui segmentent possèdent une idée différente de ces limites. Afin de déterminer l'importance de ces différences de segmentation, il serait intéressant de demander à plusieurs personnes de segmenter les images de notre base, puis d'analyser les différences entre les masques de feu et de fumée. Le coefficient de Kappa Cohen [117] serait un indicateur très intéressant pour réaliser les différences de segmentations. Cet indicateur est un test paramétrique généralement utilisé dans le milieu médical pour déterminer l'accord entre deux ou plusieurs observateurs. Dans notre cas, chaque pixel correspond à un patient et chaque personne qui segmente les images à un expert. Cette étude permettrait d'estimer l'erreur de subjectivité de segmentation. Nous pourrions également entraîner le réseau avec une des bases de segmentation ainsi réalisées et observer si le réseau diminue ou réduit l'agrément entre les différentes segmentations.

Nous avons monté dans un récent article [23] que la performance de segmentation dépendait fortement de la base de données d'entraînement. Le réseau est la plupart du temps conçu et ajusté sur la base d'entraînement, limitant la généralisation à d'autres données. La création d'une base de données de segmentation mise à disposition de la communauté internationale permettrait de comparer les performances de segmentation des divers réseaux.

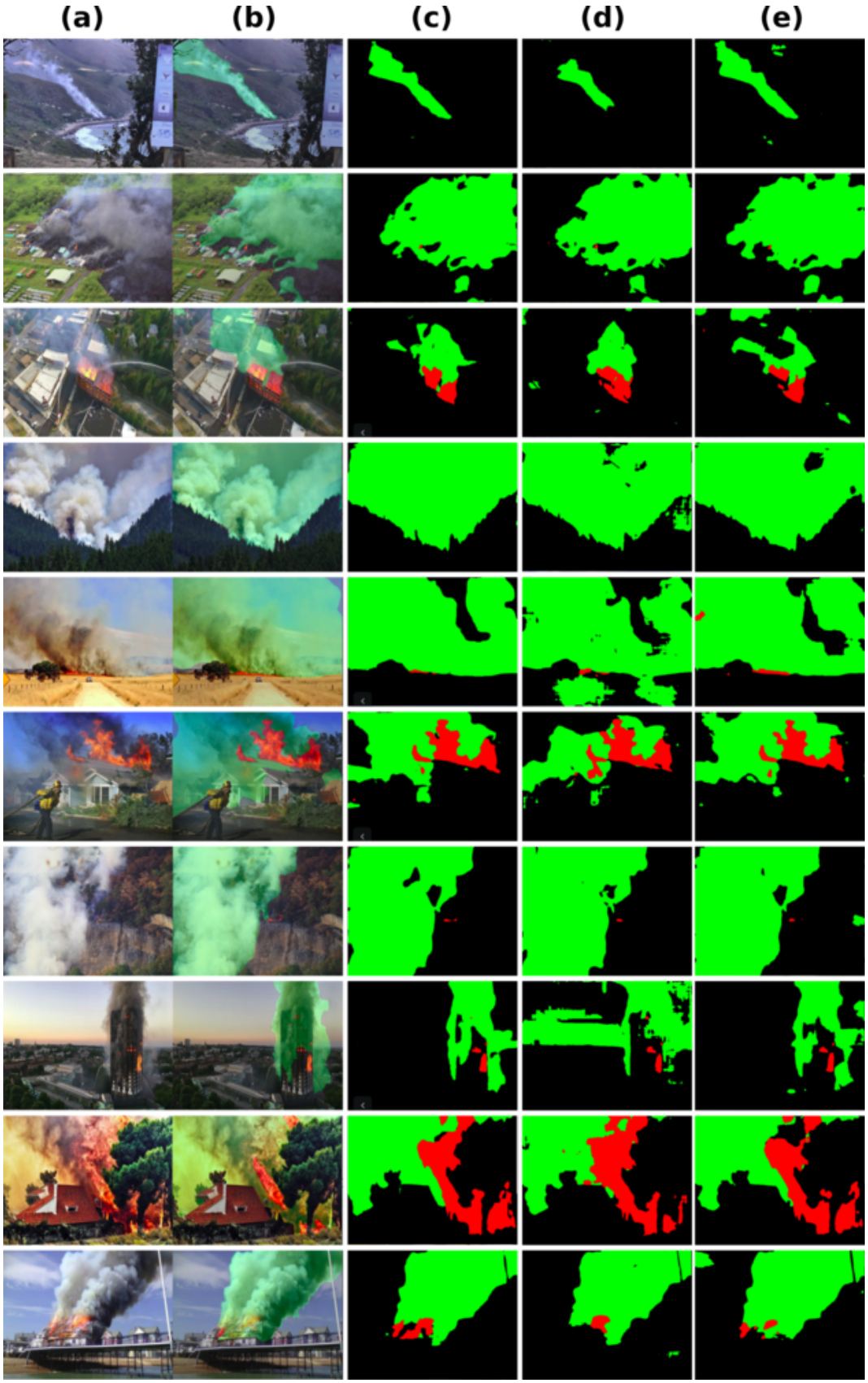


FIGURE 4.17 – *Masque de prédiction du feu et de la fumée pour les 3 réseaux (a) image originale RGB. (b) Surimposition du masque de fumée en vert et en rouge du masque de feu sur l'image d'entrée. (c)Notre réseau (Vert fumée et Rouge feu). (d) Réseau U-Net. (e) Réseau Yuan*

Chapitre 5: Conclusion générale et perspectives

Contents

Une conséquence du réchauffement climatique se matérialise par la multiplication et l'intensité des feux de forêt. Ces feux engendrent à leur tour une émission conséquente de gaz à effet de serre tel que le dioxyde de carbone et le méthane induisant un cercle vicieux par l'augmentation de la température globale de notre planète. La prévention des incendies doit être une priorité pour stopper cette spirale infernale afin de préserver la diversité de nos écosystèmes, la sécurité des populations et des biens. Les hommes et femmes politiques de la communauté européenne et mondiale sont conscients de ce problème et de son évolution. Des programmes de prévention de lutte contre les incendies émergent des instances politiques. Je me suis attelé dans mon travail de thèse à apporter une contribution à la détection et la localisation du feu et de la fumée dans des images du visible (RGB), en espérant que ce modeste apport aura un effet bénéfique sur la qualité future de la vie de nos enfants. Dans cette section, nous rappellerons les principales contributions dans le domaine de l'apprentissage profond appliqué à la détection du feu et de la fumée. Nous présenterons, pour finir des idées de pistes futures de recherche tant pour la prévention que l'aide à la lutte contre les incendies.

5.1 Récapitulatif des contributions

Dans cette thèse, nous nous sommes intéressés dans un premier temps à la détection du feu et de la fumée dans des images visibles et plus particulièrement le système de couleur RGB. L'apprentissage profond permet principalement de nous libérer de l'obligation de créer un vecteur de caractéristiques spécifiques aux objets feu et fumée. Par un apprentissage supervisé, le réseau va trouver une structure, parfois invisible à l'humain, permettant de discriminer l'arrière plan et les objets à détecter. Nous avons créé une structure légère de réseau basée sur des opérations de convolution permettant de détecter le feu et la fumée avec une grande précision. En utilisant une fenêtre glissante sur la dernière carte de caractéristiques, nous avons pu réaliser les masques de segmentation du feu et de la fumée sur des images et des vidéos.

Dans un second temps, nous nous sommes intéressés à la segmentation de précision du feu et de la fumée dans des images RGB. Pour ce faire, nous avons imaginé une structure de réseau composé d'un chemin de codage basé sur une architecture VGG16 et d'un chemin de décodage particulier. La dernière opération du chemin de codage ainsi que la structure du chemin de décodage a permis d'obtenir de très bons résultats tant sur les masques de fumée que de feu. La performance du réseau semble provenir de la grande taille du champ

réceptif ainsi que de la profondeur du réseau. Enfin, notre réseau est capable de segmenter le feu et la fumée, à l'aide d'une carte graphique commerciale utilisée par de joueur de jeux vidéos, dans des temps compatibles avec le temps réel.

La base de données, dans le cadre de l'apprentissage profond, est très importante. Elle doit refléter le mieux possible la réalité pour une bonne généralisation ultérieure de la prédiction du réseau. La quantité est un élément important ainsi que la richesse des données, d'autant que le feu et la fumée sont des objets ne possédant pas de forme propre. La détection et la segmentation du feu et de la fumée ont nécessité la réalisation manuelle de bases de données. Des applications sous python ont été réalisées. Pour la création des masques de la base d'entraînement, nous avons manuellement segmenté à l'aide du logiciel LabelMe des centaines d'images issues du net. Au-delà des données d'apprentissages, les hyper-paramètres ont été réglés suite à de nombreux essais, pour obtenir une précision de détection ou de segmentation maximale sur des images non vues par le réseau dans la phase d'apprentissage.

Plusieurs pistes de travaux futurs ont été mises en évidence lors de cette thèse. Nous allons décrire dans les sections suivant quelques-unes tant dans le domaine de la prévention que de l'aide à la lutte contre les incendies.

5.2 Perspectives

Comme nous l'avons vu précédemment dans ce mémoire, les réseaux convolutifs possèdent une architecture capable de détecter et de localiser le feu et la fumée à partir d'images RGB avec une excellente précision, tout en étant compatible avec du temps réel. La fumée est un objet difficile à détecter et segmenter de par son manque marqué de forme et sa grande variabilité colorimétrique. Dans ce cadre, les réseaux convolutifs sont parfaitement adaptés à la détection de la fumée. Toutefois, la qualité et la diversité de la base d'entraînement revêt une importance primordiale. Le souci principal actuellement réside dans le fait qu'il n'existe pas de base de données d'images, mise à la disposition de la communauté scientifique, permettant de comparer et d'évaluer les performances des divers réseaux. Les articles de recherche évaluent les performances de leur réseau sur leur base construite pour l'occasion. Les architectures des réseaux et des hyper-paramètres sont ajustées à cette base et restent la plupart du temps difficilement généralisables à d'autres bases. L'agrégation des quelques bases disponibles permettrait de régler ce problème. De plus, la création des bases de segmentations est fastidieuse à réaliser, comme nous l'avons vu précédemment dans ce mémoire, pouvant amener une certaine subjectivité lors de la réalisation des masques de fumée et de feu.

La réalisation d'une base de segmentation internationale serait la bienvenue. Outre les masques de feu et de fumée, elle pourrait contenir des masques d'objets apportant une aide aux pompiers tant sur la prévention que dans l'intervention contre les incendies comme le feu, la fumée, les habitations, les infrastructures, les routes, les points d'eau..

La fumée et le feu sont des objets qui possèdent une dynamique marquée. Dans le cadre

de caméras fixes, la prise en compte de la dimension temps (plusieurs images successives de la scène) engendrerait la mise en place d'opérations de convolutions en 3 dimensions dans le réseau. Cette troisième dimension prendrait en compte la dynamique particulière de la fumée et du feu en augmentant la précision des masques tout en réduisant le taux de faux positifs et de faux négatifs lors de présence de nuages ou la brume dans la vidéo.

La politique de lutte contre les incendies revêt deux grands objectifs : la prévention comprenant la surveillance des espaces et le management des forêts ainsi que l'efficacité de la lutte contre les incendies. Nous exposons ci-dessous les pistes de recherche potentielle pour ces deux objectifs.

5.2.1 Prévention

L'élément essentiel de lutte contre les incendies reste la prévention. Elle doit passer par une détection et une localisation le plus rapidement possible des départs de feu.

L'arrivée de la 5G va permettre d'augmenter la mise en place de caméra de vidéo-surveillance. Une application utile de cette augmentation serait de disposer sur les points culminants des caméras couvrant une zone géographique forestière par exemple *Figure 5.1*. Ces vidéos seraient envoyées vers un central disposant d'un calculateur produisant une segmentation ou une détection de la fumée. Par triangulation sur l'analyse des images de ces diverses caméras, une localisation du départ de feu serait possible.

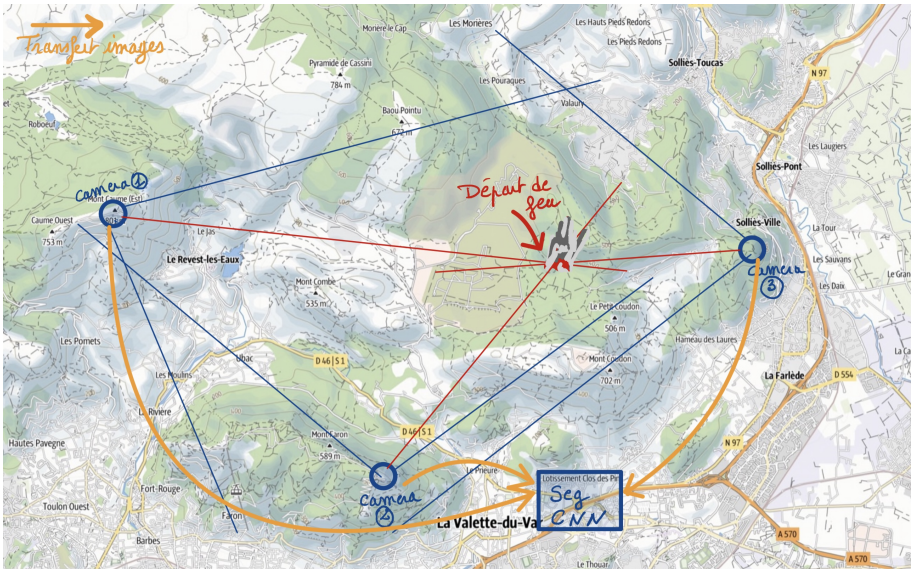


FIGURE 5.1 – *Couverture géographique par des caméras fixes - Triangulation pour la localisation du départ de feu*

Enfin, l'ultime outil de prévention serait de pouvoir déterminer la probabilité d'apparition d'un incendie sur une fenêtre de temps donnée [118], [119], [120] et [121]. Cette prédiction permettrait d'organiser les forces d'intervention et de surveillance sur des sites réduits dans le but de pouvoir intervenir le plus rapidement possible. De nos jours, les données satellitaires permettent sur des images à haute définition de créer des cartes de stress hydrique de la végétation par une analyse hyper-spectrale des émissions du couvert

végétal, d'hydrométrie, de température, de dénivelé, d'altitude, d'urbanisation (activités, routes, chemins, etc.). La collecte de toutes ces cartes dans le temps (quelques semaines ou mois avant la saison des incendies) apporterait un historique sur ces paramètres physiques.

Finalement, pour créer les masques de prédiction, nous aurions besoin de réaliser des recherches sur plusieurs années déterminant la localisation des départs de feux. Toutes ces cartes pourraient être les entrées d'un réseau convolutif produisant une carte de prédiction de départ de feu.

Cette carte de risque d'incendie pourrait servir à la gestion des espaces forestiers tant du point de vue de l'entretien que de l'urbanisation de ces lieux. Elle permettrait de développer la résilience des territoires aux grands incendies en guidant les politiques.

5.2.2 Aide à la lutte contre les incendies

Dès lors qu'un feu est déclaré, les forces d'intervention doivent agir au plus vite avec une coordination optimale. La segmentation à partir d'images dans le visible couplées à des images dans les domaines de l'infrarouge sont susceptible d'apporter des informations essentielles à la gestion de l'incendie aux pompiers. Les masques de feu et de fumée sont susceptibles de donner des informations sur l'étendue du sinistre et la direction du feu. Couplés à la connaissance du terrain et aux cartes de prédiction des risques, les pompiers pourraient anticiper la dynamique de l'incendie permettant de protéger les populations et les biens. À l'aide d'un logiciel de SIG, une superposition des infrastructures routières, des habitations ainsi que des zones touchées par le feu et/ou la fumée permettrait de planifier l'intervention et de protéger les pompiers sur le terrain.

Enfin, afin de mieux appréhender le comportement des incendies, une piste intéressante de recherche future serait de réussir à modéliser la fumée en 3D.

Bibliographie

[1] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboğlu, B. U. Töreyin, and S. Verstockt, "Video fire detection–review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, 2013.

[2] C. Yu, Z. Mei, and X. Zhang, "A real-time video fire flame and smoke detection algorithm," *Procedia Engineering*, vol. 62, pp. 891–898, 2013.

[3] B. U. Töreyin, "Fire detection algorithms using multimodal signal and image analysis," Ph.D. dissertation, bIlkent university, 2009.

[4] B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "Contour based smoke detection in video using wavelets," in *2006 14th European Signal Processing Conference*. IEEE, 2006, pp. 1–5.

[5] W. Mbarki, M. Bouchouicha, S. Frizzi, F. Tshibasu, L. B. Farhat, and M. Sayadi, "Lumbar spine discs classification based on deep convolutional neural networks using axial view mri," *Interdisciplinary Neurosurgery*, vol. 22, p. 100837, 2020.

[6] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet : A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[9] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE, 2018, pp. 117–122.

[10] G. Kang, X. Dong, L. Zheng, and Y. Yang, "Patchshuffle regularization," *arXiv preprint arXiv :1707.07103*, 2017.

[11] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 001–13 008.

[12] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks : An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[13] N. Cui, "Applying gradient descent in convolutional neural networks," in *Journal of Physics : Conference Series*, vol. 1004, no. 1. IOP Publishing, 2018, p. 012027.

[14] J. R. Marlon, P. J. Bartlein, C. Carcaillet, D. G. Gavin, S. P. Harrison, P. E. Higuera, F. Joos, M. Power, and I. Prentice, "Climate and human influences on global biomass burning over the past two millennia," *Nature Geoscience*, vol. 1, no. 10, p. 697, 2008.

[15] Z. Wang, J. Chappellaz, K. Park, and J. Mak, "Large variations in southern hemisphere biomass burning during the last 650 years," *Science*, vol. 330, no. 6011, pp. 1663–1666, 2010.

[16] Data. (2020) Viirs active fires data. [Online]. Available : https://data.globalforestwatch.org/datasets/viirs-active-fires

[17] V. R. Vallejo Calzada, N. Faivre, F. M. Cardoso Castro Rego, J. M. Moreno Rodríguez, and G. Xanthopoulos, "Forest fires. sparking firesmart policies in the eu," 2018.

[18] J.-C. Ciscar, L. Feyen, A. Soria, C. Lavalle, F. Raes, M. Perry, F. Nemry, H. Demirel, M. Rozsai, A. Dosio *et al.*, "Climate impacts in europe-the jrc peseta ii project," 2014.

[19] P. Barbosa, J. Kucera, P. Strobl, P. Vogt, A. Camia, and J. s. San-Miguel-Ayanz, "European forest fire information system (effis) ? rapid damage assessment : Appraisal of burnt area maps in southern europe using modis data (2003 ? 2005)," *Forest Ecology and Management*, vol. 234, no. 1, p. S218, 2006.

[20] État. (1973) Prometheebase de données. [Online]. Available : http://www.promethee.com

[21] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 877–882.

[22] S. Frizzi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and M. Sayadi, "Convolutional neural network for smoke and fire semantic segmentation," *IET Image Processing*, vol. 15, no. 3, pp. 634–647, 2021.

[23] S. Frizzi, M. Bouchouicha, and E. Moreau, "Comparison of two semantic segmentation databases for smoke detection," in *IEEE International Conference on Industrial Technology*, 2021.

[24] R. Kaabi, S. Frizzi, M. Bouchouicha, F. Fnaiech, and E. Moreau, "Video smoke detection review : State of the art of smoke detection in visible and ir range," in *2017 International Conference on Smart, Monitored and Controlled Cities (SM2C)*. IEEE, 2017, pp. 81–86.

[25] W. Mbarki, M. Bouchouicha, S. Frizzi, F. Tshibasu, L. B. Farhat, and M. Sayadi, "A novel method based on deep learning for herniated lumbar disc segmentation," in *2020 4th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*. IEEE, 2020, pp. 394–399.

[26] A. Laks and M. Rashed, *Aristote et le mouvement des animaux : Dix études sur le De motu animalium*. Presses Univ. Septentrion, 2004, vol. 886.

[27] F. Gall, "Préface a fj gall y g. spurzheim," *Anatomie et Physiologie du système nerveux en général et du cerveau en particulier, avec des observations sur la possibilité de reconnoître plusieurs dispositions intellectuelles et morales de l'homme et des animaux, par la configuration de leurs têtes*, pp. 229–240, 1810.

[28] T. Schwann, *Microscopical Researches into the Accordance in the Structure and Growth of Animals and Plants*, 1847.

[29] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[30] D. Andler, "Connexionnisme et cognition : à la recherche des bonnes questions," *Revue de synthèse*, vol. 111, no. 1, pp. 95–127, 1990.

[31] D. Hebb, "0.(1949)," *The organization of behavior*, 1957.

[32] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," CORNELL AERONAUTICAL LAB INC BUFFALO NY, Tech. Rep., 1961.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[34] B. Regression, "New tools for prediction and analysis in the behavioral sciences," *Werbos, PJ https ://books. google. se/books*, 1974.

[35] Y. Le Cun, "Une procedure d'apprentissage pour reseau a seuil assymetrique [a learning procedure for asymmetric threshold network], proceedings of cognitiva 85, 599-604," 1985.

[36] D. B. Parker, "Learning logic," 1985.

[37] M. L. Minski and S. A. Papert, "Perceptrons : an introduction to computational geometry," *MA : MIT Press, Cambridge*, 1969.

[38] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in *2004 International Conference on Image Processing, 2004. ICIP'04.*, vol. 3. IEEE, 2004, pp. 1707–1710.

[39] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder : Real-time tracking of the human body," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[40] Q. Zhou and J. K. Aggarwal, "Tracking and classifying moving objects from video," in *Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, vol. 12. Hawaii, USA, 2001.

[41] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 246–252.

[42] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.

[43] A. Shimada, D. Arita, and R.-i. Taniguchi, "Dynamic control of adaptive mixture-of-gaussians background model," in *2006 IEEE International Conference on Video and Signal Based Surveillance*. IEEE, 2006, pp. 5–5.

[44] L. Carminati and J. Benois-Pineau, "Gaussian mixture classification for moving object detection in video surveillance environment," in *IEEE International Conference on Image Processing 2005*, vol. 3. IEEE, 2005, pp. III–113.

[45] Z. Xu and J. Xu, "Automatic fire smoke detection based on image visual features," in *2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*. IEEE, 2007, pp. 316–319.

[46] P. Piccinini, S. Calderara, and R. Cucchiara, "Reliable smoke detection in the domains of image energy and color," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 1376–1379.

[47] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern recognition letters*, vol. 27, no. 1, pp. 49–58, 2006.

[48] T. Celik, H. Ozkaramanlt, and H. Demirel, "Fire pixel classification using fuzzy logic and statistical color model," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 1. IEEE, 2007, pp. I–1205.

[49] S. Calderara, P. Piccinini, and R. Cucchiara, "Smoke detection in video surveillance : a mog model in the wavelet domain," in *International conference on computer vision systems*. Springer, 2008, pp. 119–128.

[50] F. Yuan, "A fast accumulative motion orientation model based on integral image for video smoke detection," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 925–932, 2008.

[51] J. Chen, Y. He, and J. Wang, "Multi-feature fusion based fast video flame detection," *Building and Environment*, vol. 45, no. 5, pp. 1113–1122, 2010.

[52] B. Ko, K.-H. Cheong, and J.-Y. Nam, "Early fire detection algorithm based on irregular patterns of flames and hierarchical bayesian networks," *Fire safety journal*, vol. 45, no. 4, pp. 262–270, 2010.

[53] T. Celik, "Fast and efficient method for fire detection using image processing," *ETRI journal*, vol. 32, no. 6, pp. 881–890, 2010.

[54] B. K. Horn and B. G. Schunck, "Determining optical flow," in *Techniques and Applications of Image Understanding*, vol. 281. International Society for Optics and Photonics, 1981, pp. 319–331.

[55] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[56] S. Harlapur and D. K. Nataraj, "Fire detection using optical flow method in videos," *Int. J. Eng. Res. Technol. IJERT*, vol. 4, no. 05, 2015.

[57] M. Mueller, P. Karasev, I. Kolesov, and A. Tannenbaum, "Optical flow estimation for flame detection in videos," *IEEE Transactions on image processing*, vol. 22, no. 7, pp. 2786–2797, 2013.

[58] G. Healey, D. Slater, T. Lin, B. Drda, and A. D. Goedeke, "A system for real-time fire detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1993, pp. 605–606.

[59] D. Rizzotti, N. Schibli, and W. Straumann, "Process and device for detecting fires based on image analysis," Aug. 30 2005, uS Patent 6,937,743.

[60] C.-B. Liu and N. Ahuja, "Vision based fire detection," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 4. IEEE, 2004, pp. 134–137.

[61] V. Gubernov, A. Kolobov, A. Polezhaev, H. Sidhu, and G. Mercer, "Period doubling and chaotic transient in a model of chain-branching combustion wave propagation," *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Sciences*, vol. 466, no. 2121, pp. 2747–2769, 2010.

[62] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt *et al.*, "A system for video surveillance and monitoring," *VSAM final report*, vol. 2000, pp. 1–68, 2000.

[63] O. Günay, K. Taşdemir, B. U. Töreyin, and A. E. Çetin, "Fire detection in video using lms based active learning," *Fire technology*, vol. 46, no. 3, pp. 551–577, 2010.

[64] B. U. Toreyin and A. E. Cetin, "Online detection of fire in video," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–5.

[65] T. Celik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire safety journal*, vol. 44, no. 2, pp. 147–158, 2009.

[66] C. S. Sherrington, "Observations on the scratch-reflex in the spinal dog," *The Journal of physiology*, vol. 34, no. 1-2, pp. 1–50, 1906.

[67] H. K. Hartline, "The receptive fields of optic nerve fibers," *American Journal of Physiology-Legacy Content*, vol. 130, no. 4, pp. 690–699, 1940.

[68] ——, "The response of single optic nerve fibers of the vertebrate eye to illumination of the retina," *American Journal of Physiology-Legacy Content*, vol. 121, no. 2, pp. 400–415, 1938.

[69] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[70] K. Fukushima and S. Miyake, "Neocognitron : A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[71] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[72] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[73] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[74] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv :1409.1556*, 2014.

[75] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[76] ——, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[77] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[78] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[80] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning : A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[81] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[82] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv :1412.7062*, 2014.

[83] J. Fu, J. Liu, Y. Wang, J. Zhou, C. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," *IEEE Transactions on Image Processing*, 2019.

[84] A. Chaurasia and E. Culurciello, "Linknet : Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[85] Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang, "Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3029–3037.

[86] O. Ronneberger, P. Fischer, and T. Brox, "U-net : Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[87] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net : Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 565–571.

[88] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *International journal of multimedia information retrieval*, vol. 7, no. 2, pp. 87–93, 2018.

[89] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, 2019.

[90] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[91] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv :1706.05587*, 2017.

[92] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[93] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[94] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv :1712.04621*, 2017.

[95] F. J. Moreno-Barea, F. Strazzera, J. M. Jerez, D. Urda, and L. Franco, "Forward noise adjustment scheme for data augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 728–734.

[96] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv :1406.2661*, 2014.

[97] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[98] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers : Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[99] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*. PMLR, 2013, pp. 1139–1147.

[100] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011.

[101] D. P. Kingma and J. Ba, "Adam : A method for stochastic optimization," *arXiv preprint arXiv :1412.6980*, 2014.

[102] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv :1609.04747*, 2016.

[103] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv :1505.00853*, 2015.

[104] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout : a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[105] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme : a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[106] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once : Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[107] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3203–3212.

[108] F. Yuan, L. Zhang, X. Xia, B. Wan, Q. Huang, and X. Li, "Deep smoke segmentation," *Neurocomputing*, vol. 357, pp. 248–260, 2019.

[109] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4159–4167.

[110] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[111] P. Isaac, "Egan, jp" signal detection theory and roc analysis"(book review)," *The Psychological Record*, vol. 26, p. 567, 1976.

[112] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[113] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.

[114] N. J. Perkins and E. F. Schisterman, "The inconsistency of "optimal" cutpoints obtained using two criteria based on the receiver operating characteristic curve," *American journal of epidemiology*, vol. 163, no. 7, pp. 670–675, 2006.

[115] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[116] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *arXiv preprint arXiv :1701.04128*, 2017.

[117] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[118] G. Zhang, M. Wang, and K. Liu, "Forest fire susceptibility modeling using a convolutional neural network for yunnan province of china," *International Journal of Disaster Risk Science*, vol. 10, no. 3, pp. 386–403, 2019.

[119] D. T. Bui, Q.-T. Bui, Q.-P. Nguyen, B. Pradhan, H. Nampak, and P. T. Trinh, "A hybrid artificial intelligence approach using gis-based neural-fuzzy inference system and particle swarm optimization for forest fire susceptibility modeling at a tropical area," *Agricultural and forest meteorology*, vol. 233, pp. 32–44, 2017.

[120] Z. S. Pourtaghi, H. R. Pourghasemi, R. Aretano, and T. Semeraro, "Investigation of general indicators influencing on forest fire and its susceptibility modeling using different data mining techniques," *Ecological indicators*, vol. 64, pp. 72–84, 2016.

[121] Q. Renard, R. Pélissier, B. Ramesh, and N. Kodandapani, "Environmental susceptibility model for predicting forest fire occurrence in the western ghats of india," *International Journal of Wildland Fire*, vol. 21, no. 4, pp. 368–379, 2012.