Reconnaissance des documents avec de l'apprentissage profond pour la réalité augmentée

Thibault Lelong

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAS017

Thèse de doctorat

Reconnaissance des documents avec de l'apprentissage profond pour la réalité augmentée

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Signal, Images, Automatique et Robotique

Thèse présentée et soutenue à Evry, le 12 Décembre 2023, par

THIBAULT LELONG

Composition du Jury :

Mihai CIUC Professeur, Polytechnic University of Bucharest (LAPI)	Président
Valeriu VRABIE Professeur, Université de Reims Champagne-Ardenne (CReSTIC)	Rapporteur
Christophe GRAVIER Professeur, Télécom Saint-Etienne	Rapporteur
Emmanuel BRICARD Ingenieur, Shift89	Examinateur
Titus ZAHARIA Professeur, Telecom SudParis (SAVOMAR)	Directeur de thèse
Marius PREDA Maitre de conferences, Telecom SudParis (SAVOMAR)	Encadrant de thèse
Michael MERLANGE CTO, ARGO SAS	Invité

Remerciements

Tout d'abord, je tiens à exprimer ma gratitude envers les différents membres de mon jury de thèse : Messieurs Valeriu VRABIE et Christophe GRAVIER, qui ont eu l'honneur d'accepter le rôle de rapporteur, ainsi que Messieurs Mihai CIUC et Emmanuel BRICARD pour leur participation en tant qu'examinateurs. Je souhaite également adresser mes remerciements à mon directeur de thèse, Titus ZAHARIA, pour ses conseils, sa réactivité et ses retours précieux, que ce soit pour la rédaction des différentes publications ou de ce manuscrit.

Je tiens également à remercier mon co-encadrant de thèse, Marius PREDA, ainsi que Michael MERLANGE, mon encadrant en entreprise, qui m'ont accompagné au quotidien tout au long de cette thèse. À travers nos discussions, ils ont généreusement partagé leurs connaissances et perspectives, contribuant ainsi à améliorer la qualité de ce travail.

Mes remerciements vont également à l'ensemble du personnel du laboratoire SAVOMAR pour leur accueil bienveillant au cours de ces trois années. Je souhaite également exprimer ma reconnaissance envers les différents salariés de la société ARGO, en particulier Guy Le HENAFF, Christophe BOSSUT et Pierre ADDOUM.

Je tiens à remercier du fond du cœur ma famille et tous mes amis pour leur soutien constant. Un merci tout spécial à mon grand frère pour sa motivation sans faille, et à ma compagne pour son soutien indéfectible tout au long de ce parcours.

Enfin, je tiens à exprimer toute ma gratitude et mon admiration pour mes parents qui m'ont soutenu, guidé et montré l'exemple, sans quoi je n'aurais pas pu atteindre le point où je me trouve aujourd'hui.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Contexte et enjeux

1.1 Contexte

Dans un monde en perpétuelle mutation technologique, l'essor remarquable des dispositifs mobiles, dotés de capacités de capture d'images sophistiquées, a repositionné l'analyse et la reconnaissance d'images comme des axes économiques incontournables. En réponse à cette tendance, de nouvelles applications ont émergé, axées sur la réalité augmentée, la localisation visuelle et la reconnaissance spécifique d'éléments variés tels que la nourriture, la flore ou les vêtements. Leur ambition première est d'augmenter la profondeur et la qualité de l'expérience utilisateur, en offrant des informations contextuelles ou des immersions basées sur une simple capture.

Toutefois, une observation minutieuse révèle un paradoxe intéressant. En dépit d'une marche rapide vers la digitalisation, notre société continue d'être profondément ancrée dans l'information tangible, comme en témoignent les vastes quantités de données toujours disponibles sous forme imprimée. Livres, magazines, et affichages publics ne sont que quelques exemples de supports traditionnels qui, malgré leur apparente obsolescence, témoignent de leur pertinence et de leur importance culturelle. Cette pérennité du support physique contraste toutefois avec l'agilité et la flexibilité des informations numériques, dont la force réside dans leur adaptabilité et leur potentialité à fusionner et à interconnecter divers contenus.

C'est dans ce contexte que les applications de réalité augmentée prennent toute leur dimension. Cherchant à combler le fossé entre le numérique et le tangible, elles intègrent des éléments numériques de manière harmonieuse dans un environnement réel. En exploitant des technologies de pointe, ces applications évaluent la pose de la caméra physique et adaptent la position et l'orientation d'une caméra virtuelle, présente dans un moteur graphique tel que Three.js, pour chaque image d'une séquence vidéo. Il est intéressant de noter que, malgré les différences apparentes dans leur mise en œuvre, les expériences en réalité augmentée peuvent être classifiées en deux catégories principales : "basées sur des marqueurs" et "sans marqueurs", témoignant de la diversité et de la complexité de ce domaine en plein essor.

Les applications de réalité augmentée dite "basées sur des marqueurs" visent à superposer des contenus numériques sur une image ou un document préalablement imprimé. On parle dans notre cas d'usage de marqueur naturel, en opposition avec les marqueurs artificiels tels que les QrCode. La première étape de cette approche consiste à détecter et à reconnaître ledit marqueur, souvent à travers un système spécialisé de reconnaissance d'images. Par la suite, la pose relative de la caméra est déterminée par rapport à ce marqueur au sein d'un flux vidéo continu.

Plusieurs entités industrielles majeures offrent des Kits de Développement Logiciel (SDK) ou des plateformes dédiées à la conception d'expériences et d'applications de cette nature. Parmi ces entités, on peut citer Vuforia, Blippard, 8th Wall et Onirix. La figure 1.1 illustre un exemple d'application Web basée sur un marqueur naturel proposée par là société ARGO.

Quant aux applications de réalité augmentée "sans marqueurs", elles visent l'incorporation fluide d'éléments virtuels dans l'environnement réel. Cette approche nécessite donc l'évaluation précise de la position de la caméra en fonction des informations environnantes, par exemple grâce à des techniques d'odométrie visuelle [LM13; Leu+15]. Dans ce domaine, plusieurs SDK ont vu le jour, avec des acteurs dominants comme Google ARCore et Apple ARKit, exploitant les atouts d'une caméra monoculaire en association avec les données d'une Unité de Mesure Inertielle (IMU).

En résumé, la principale distinction entre les applications de réalité augmentée "basées sur des

FIGURE 1.1 – Exemple d'une application de réalité augmentée basée sur un marqueur naturel, utilisant le moteur de suivi d'images développé pour les applications Web. Pour lancer l'application Web, il faut scanner le QR Code fourni pour être redirigé vers une application Web. Une fois l'application lancée, il suffit de scanner simplement l'image à droite du QrCode pour découvrir l'expérience de réalité augmentée.

marqueurs" et "sans marqueurs" réside dans l'estimation de la pose de la caméra qui peut être soit intrinsèquement liée à une image ou un document, soit être déterminée en fonction de l'environnement global.

Dans ce contexte, la société ARGO a pour objectif et enjeux d'instaurer une passerelle robuste entre un support physique et sa réplique numérique, en exploitant les potentialités offertes par les applications de réalité augmentée basée sur marqueur naturel. En conséquence, les deux axes de recherches que nous poursuivrons sont les suivants :

— Identification d'un marqueur naturel présent au sein d'une base de données depuis une caméra,
— Suivi en temps réel d'un marqueur depuis une caméra pour estimer sa position.

1.2 Les limites et les enjeux

Les documents imprimés revêtent une diversité informationnelle notable incluant à la fois des éléments graphiques et textuels. D'une part, les éléments graphiques, dans leur essence, englobent un éventail de représentations, de la photographie aux emblèmes, incluant également dessins, affiches et œuvres

picturales. Ces composants visuels ont l'avantage de transmettre des informations de manière succincte et parfois plus efficace que les contenus textuels, facilitant ainsi l'identification des médias physiques associés.

D'autre part, les éléments textuels incarnent des vecteurs d'information potentiellement distinctifs, conférant l'aptitude d'identifier un document avec précision au sein d'une collection. Néanmoins, l'extraction de ces données textuelles peut rencontrer des défis non négligeables, dépendant du contexte.

L'exploration et l'identification de documents via des dispositifs de capture, tels que les caméras, se positionne donc à la frontière entre la Recherche d'Informations (IR - Information Retrieval) et la Recherche d'Images Basée sur le Contenu (CBIR - Content-Based Image Retrieval). La finalité intrinsèque de cette tâche, à savoir la reconnaissance de documents, réside dans l'identification d'une image au sein d'une base de données, à partir d'une requête visuelle de l'utilisateur, en évitant toute ambiguïté.

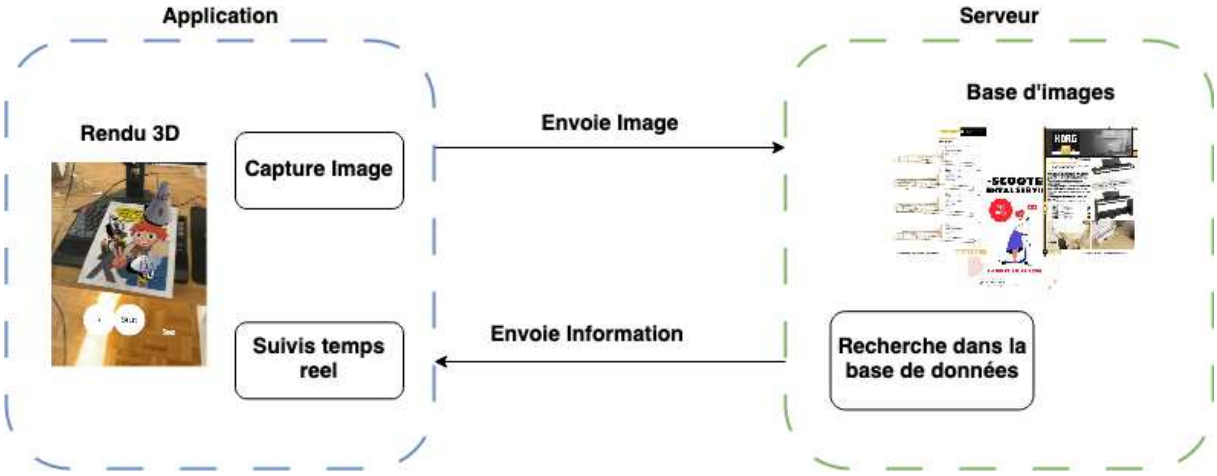


FIGURE 1.2 – Architecture classique d'une application de réalité augmentée

Ainsi, les mécanismes de récupération d'images ou de documents s'articulent autour de plusieurs phases. Initialement, lors d'une étape hors ligne, les images de référence sont intégrées au serveur. Ultérieurement, dans la phase dite en ligne, une image est capturée via un dispositif utilisateur puis transmise au serveur afin d'orchestrer une recherche au sein de la base de données. Néanmoins, la détection d'images au sein d'une base de données se heurte à diverses contraintes :

- La résolution souvent modeste des images obtenues via des caméras.
- Une maîtrise limitée des conditions d'éclairage par la caméra comparativement à un scanner à plat, induisant des variations lumineuses attribuables à l'environnement et au dispositif.
- L'éventuelle présence de distorsions de perspective en raison de la mobilité du dispositif de capture.
- La volatilité intrinsèque des appareils mobiles implique que le dispositif ou la cible pourrait être en mouvement, générant des altérations telles que le flou.
- De surcroît, la capture via caméra tend à extraire un fragment de l'image de référence, impliquant une concordance partielle entre l'image obtenue et l'originale, d'où la nécessité de concevoir une méthodologie efficiente pour apparier les images de documents.

Devant ces défis inhérents à l'identification, la reconnaissance et le suivi d'images, deux approches se distinguent nettement : la reconnaissance Optique de Caractère (OCR) et la reconnaissance basée

sur les caractéristiques visuelles.

1.2.1 Reconnaissance optique de caractères

La méthode de Reconnaissance Optique de Caractères (OCR) s'établit comme une solution privilégiée pour la gestion des images composées majoritairement d'éléments textuels. Elle consiste à transcrire les images, qu'elles soient de référence ou issues de requêtes en format textuel. Ce processus requiert une série de pré-traitements visant à extraire termes et segments textuels des images. Les résultats de cette transcription sont ensuite indexés au sein d'une base de données ou utilisés comme critères de requêtes, en vue de l'identification de documents similaires. L'efficacité de l'OCR est optimale lorsque les documents sont capturés dans des conditions rigoureuses, notamment, lorsque les images proviennent de scans et que le contraste entre le texte et son arrière-plan est nettement perceptible.

Toutefois, cette approche présente des limites notables :

— La dépendance linguistique des contenus documentaires,
— La résolution des images, qu'elles soient de référence ou de requête,
— La prédominance d'éléments graphiques ou leur caractère discriminant,
— La sensibilité aux distorsions géométriques telles que le pliage, les vues en perspective, etc,
— La vulnérabilité face aux distorsions photométriques, notamment celles liées à des conditions d'éclairage variables, aux mouvements, aux ombres, ou encore aux captures partielles.

1.2.2 Reconnaissance par caractéristique visuelle

Les caractéristiques visuelles, quant à elles, sont des éléments spécifiques et pertinents issus d'une image, utilisés pour en faciliter la reconnaissance, la classification ou l'analyse. Ces éléments englobent des points d'intérêt distinctifs, des contours délimitant la morphologie d'un objet ou encore la texture de l'image. Les nuances colorimétriques, avec leurs intensités et distributions, constituent, également, des repères primordiaux. Des techniques sophistiquées, telles que les descripteurs locaux (SIFT, SURF), offrent une analyse approfondie des zones d'intérêt en résistant aux variations d'échelle, de rotation ou d'éclairage.

Avec l'émergence de l'apprentissage profond, les réseaux neuronaux convolutifs peuvent également détecter des caractéristiques visuelles de manière hiérarchisée, rendant la reconnaissance d'images nettement plus performante. Ces éléments sont essentiels en vision par ordinateur, car ils facilitent l'interprétation visuelle par les machines.

En résumé, la méthode OCR, compte tenu des limitations précédemment évoquées, ne semble pas être la plus adaptée aux problématiques soulevées pour la reconnaissance, l'identification et le suivi d'image depuis une caméra. Nos travaux se focaliseront donc essentiellement sur l'étude et l'utilisation de caractéristiques visuelles pour les deux axes de recherche précédemment cités.

Les présentes recherches ont bénéficié d'un financement au titre d'une collaboration CIFRE entre Télécom SudParis et la société ARGO. L'orientation prédominante de ces investigations se caractérise par une visée industrielle, se manifestant dans le développement de systèmes intégrés et fonctionnels, destinés à la reconnaissance et au suivi d'images, dans le but ultime de générer des expériences de réalité augmentée sur navigateur Web et de doter la société ARGO d'une technologie propriétaire.

1.3 Objectifs et contributions

Au sein de cette thèse, les objectifs et contributions peuvent être classifiés en deux principales catégories, chacune avec d'importantes implications industrielles.

La première catégorie englobe la création d'un moteur de reconnaissance d'images visant à se rapprocher d'un système d'identification. La société ARGO s'appuie actuellement sur la solution fournie par Vuforia pour la reconnaissance d'images. Toutefois, cette solution présente des insuffisances, notamment en termes de confusion entre des images partageant des similitudes dans la base de données lors de requêtes. Par ailleurs, certaines images, grâce à leur contenu textuel prédominant et leur répartition hétérogène, posent des défis d'identification.

La seconde catégorie d'objectifs s'oriente vers la création d'une bibliothèque de traitement d'images spécifiquement optimisée pour un usage Web. Cela revient à établir un moteur de suivi d'images. Face à la tendance croissante du développement d'applications Web, il est impératif pour ARGO de disposer et de fournir sa propre solution Web. Créer un moteur de suivi d'images pour des environnements Web impose des contraintes, notamment une capacité de calcul restreinte et une nécessité de minimiser la taille de la bibliothèque à charger. L'ambition majeure est de proposer une expérience de réalité augmentée via une application Web, rivalisant ainsi avec une application native (application qui nécessite une installation au préalable sur une plateforme ou un système d'exploitation particulier, comme iOS pour les appareils Apple ou Android), tout en demeurant fonctionnelle sur des dispositifs de gamme intermédiaire.

Dans ce cadre, nos principales contributions sont les suivantes :

— L'élaboration d'une base de données s'alignant sur une vérité terrain de nature industrielle.
— La conception et la mise en œuvre d'un modèle de détection de documents adapté aux applications Web.
— La conception d'un pipeline pour la reconnaissance d'images.
— La création d'un moteur de suivi d'images, à la fois compact et temps réel, destiné aux applications Web.

1.4 Organisation du manuscrit

Le reste de ce manuscrit est structuré comme suit :

— Le chapitre chapitre 2 propose une revue de l'état de l'art des différentes méthodes permettant de reconnaître et identifier un document à partir d'une image naturelle dans une base de données.
— Le chapitre chapitre 3 présente en détail la nouvelle base de données que nous avons constituée, ainsi qu'un banc de tests permettant d'évaluer les principales approches de l'état de l'art. Seront également présentées les différentes bases de données disponibles en lien avec notre sujet de recherches, ainsi que leurs limitations.
— Le chapitre chapitre 4 introduit le modèle de détection de documents proposé, ainsi qu'un nouveau pipeline spécialisé dans la reconnaissance d'image.
— Le chapitre chapitre 5 se concentre sur nos travaux axés sur la production industrielle, avec notre bibliothèque permettant de construire un moteur de suivi d'images en temps réel pour les applications Web. Ce système est actuellement utilisé par la société ARGO, et commercialisé.

Chapitre 2

État de l'art

Ce chapitre explore les méthodes de création de descripteurs, qu'ils soient locaux ou globaux, utilisées pour caractériser des informations dans des images. On y aborde, dans un premier temps, les méthodes traditionnelles basées sur la description, comme SIFT, SURF, qui identifient et décrivent des caractéristiques spécifiques à partir d'une analyse détaillée de l'image. Puis, on se penche sur les méthodes basées sur l'apprentissage, notamment avec les réseaux de neurones profonds, qui permettent d'extraire des caractéristiques en s'entraînant sur d'importants ensembles de données.

2.1 Introduction

Les applications de réalité augmentée incluent en général deux parties distinctes. La première partie concerne la reconnaissance d'images ou de documents. Ces moteurs de recherche d'images doivent être capables de gérer des images contenant à la fois du texte et des éléments graphiques. Ils utilisent deux types d'approche :

— détection et analyse des caractères présents dans l'image (OCR) ;
— identification d'un document similaire dans une base de données (même concept que les moteurs de recherche d'images).

La seconde partie concerne les systèmes de suivi d'images en temps réel. De nombreux travaux ont été réalisés, afin d'estimer la position d'une image plane en temps réel, dont l'objectif peut être de concevoir des applications de réalité augmentée. Nous retrouvons d'ailleurs des méthodes similaires au système de reconnaissance d'images que nous détaillerons par la suite.

La tendance actuelle est cependant au passage des applications natives vers le Web, ce qui implique un certain nombre de contraintes [Qia+19] telles que des limitations de puissance de calcul, limitation du poids des fichiers sources à charger à chaque ouverture de l'application Web, etc.

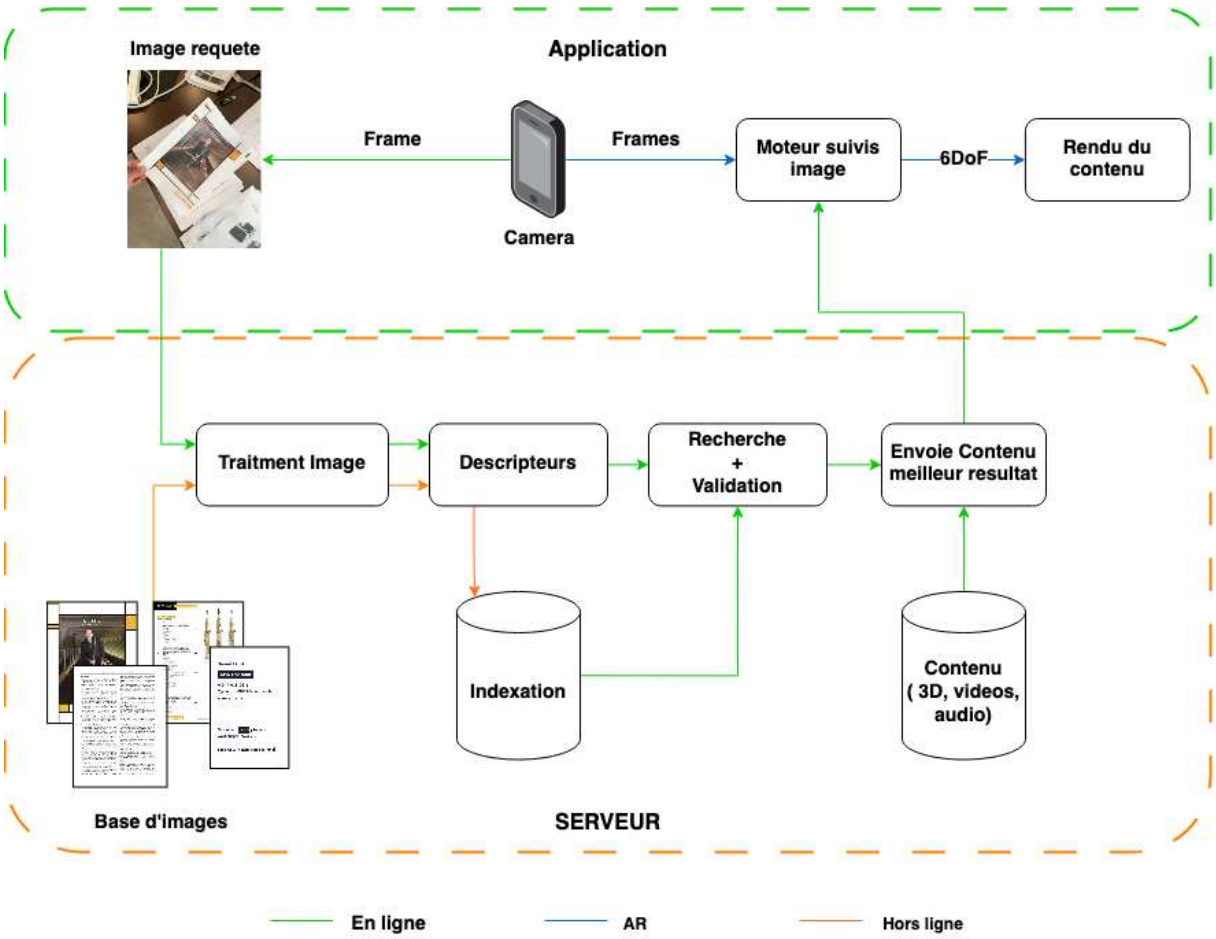


FIGURE 2.1 – Représentation d'un pipeline générique de reconnaissance d'images/documents pour les applications de réalité augmentées ou autres

Comme illustré figure 2.1, le pipeline d'une application de réalité augmentée à base de marqueur peut se décomposer en trois étapes de traitement distinctes :

— Mode hors ligne : cette étape consiste à extraire un ou plusieurs vecteurs de caractéristiques (descripteurs), qui décrivent les informations visuelles contenues dans une image. Les descripteurs provenant d'un ensemble d'images qui forment une base de données de référence sont ensuite stockés et indexés.
— Mode en ligne : cette étape représente la recherche et l'identification de l'image à partir d'une photographie capturée par l'utilisateur, représentant la requête. Après extraction des mêmes vecteurs de caractéristiques une mesure de similarité est utilisée pour déterminer l'image correspondante dans la base de données de référence et renvoyer par la suite les différents contenus 3D, audio ou vidéo à l'application liée à cette image.
— Mode Réalité Augmentée : cette étape démarre une fois qu'un document a été identifié par le système de reconnaissance. L'application va alors charger l'ensemble des contenus relatifs à l'image identifiée. Puis, un moteur de suivi d'images va suivre en temps réel le marqueur de référence, afin d'estimer son emplacement, permettant ainsi de positionner de manière cohérente les différents contenus.

Ces différents traitement nécessite dans leur ensemble la spécification de descripteurs visuels adaptés et discriminants, qu'ils soient obtenus à l'aide de méthodes traditionnelles ou en utilisant des techniques d'apprentissage profond.

Deux grands types de descripteurts visuels peuvent être utilisés : globaux ou locaux.

Les descripteurs locaux sont calculés sur des régions spécifiques définies autour d'un ensemble de "points d'intérêt" ou "points clés". Ils représentent de manière concise l'information contenue dans ces zones en utilisant des caractéristiques telles que l'apparence, la texture, la forme ou le voisinage. A contrario, les descripteurs globaux ne décrivent pas de régions spécifiques, mais sont calculés sur l'ensemble de l'image. Des méthodes d'encodage et de quantification des descripteurs locaux sont également utilisées pour construire des représentations uniques et plus compactes à partir d'un ensemble de données. Mentionnons également les méthodes d'agrégation qui rendent possible de globaliser l'information portée par l'ensemble des descripteurs locaux associés à une image donnée dans un vecteur de description global, décrivant l'image dans sa totalité.

Les différents descripteurs sont ensuite utilisés pour apparier les caractéristiques détectées entre différentes images. Cette étape est réalisée dans l'espace des descripteurs, à l'aide de mesures de similarité et/ou distances dédiées. L'objectif est de trouver les paires de caractéristiques correspondantes entre les images afin de d'estimer les régions ou les images similaires.

Notons qu'il est également possible de localiser des régions dans une image, à l'aide de techniques de vérification/validation géométrique, ce qui est essentiel pour les applications de réalité augmentée à base de marqueurs.

Dans un premier temps, nous présenterons les différentes méthodes de construction de descripteurs, que ce soit par description ou par apprentissage, puis nous aborderons les différentes étapes de prétraitement relatives à la reconnaissance d'images lorsqu'il s'agit de documents. Dans notre cas, ces traitements englobent les systèmes de segmentation, de détection et de correction des images/documents qui peuvent avoir un impact significatif sur la précision du système.

2.2 Descripteurs visuels d'image

L'objectif de tout descripteur visuel est de capturer au mieux les caractéristiques visuelles d'une image. Afin d'être discriminants et fonctionnels, les descripteurs doivent satisfaire un certain nombre de propriétés :

— Invariance aux transformations géométriques de similarité : les descripteurs doivent être similaires, quelle que soit la position, l'échelle et l'orientation de l'image ou de la région que l'on souhaite décrire.
— Invariance aux changements d'éclairage : les descripteurs doivent être robustes par rapport aux variations de luminosité ou de contraste de l'image.
— Répétabilité : les descripteurs doivent être similaires dans différentes images d'une même scène, notamment lorsque la pose de la caméra est différente.
— Distinctivité : les descripteurs doivent être uniques et facilement identifiables, même dans des images bruyantes ou dans le cas des régions d'image peu texturées.

Dans la riche litérature dédiée à ce sujet, nous pouvons distinguer trois grands types de descripteurs : globaux, locaux et par agrégation.

2.2.1 Descripteurs globaux

Les descripteurs globaux ont pour objectif de fournir, à partir d'une image I, un vecteur de caractéristiques de dimension L (Figure 2.2) contenant les propriétés de l'image telles que la forme, les contours, la couleur ou encore la texture.

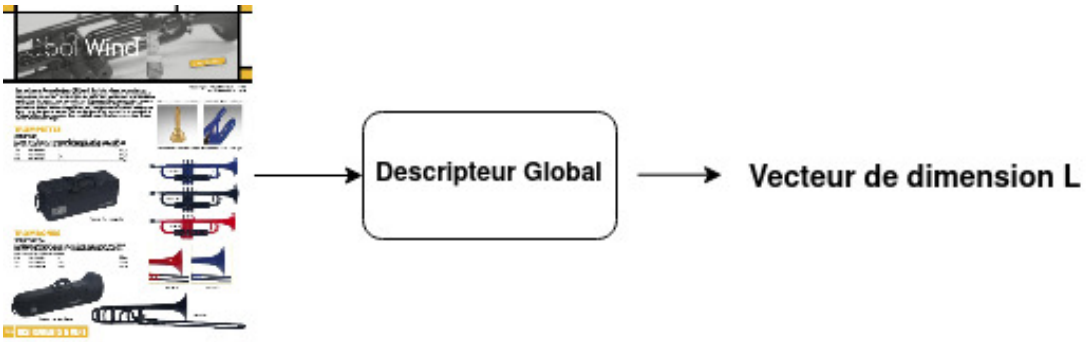


FIGURE 2.2 – Construction d'un descripteur global

Une des techniques les plus classiques et pionnière dans le domaine de l'indexation par le contenu, s'appuie sur l'utilisation d'histogrammes de couleur [SB91]. Chaque image est représentée par un histogramme de couleur, construit dans un espace de couleurs fortement quantifié. Une intersection d'histogrammes est ensuite utilisée comme mesure de similarité. De manière générale, les histogrammes offrent l'avantage de la robustesse aux changements de résolution et aux variations d'angles de vue. Cependant, cette approche demeure à l'évidence bien trop simpliste pour prendre en compte la richesse informationelle présente dans une image.

Depuis, un nombre impressionnant de travaux de recherche à l'échelle mondiale a permis l'essor du domaine de l'indexation par le contenu. Dans ce cadre, mentionnons la sortie au début des années 2001

de la norme ISO/MPEG-7 [Pas+12], qui proposait un ensemble complet de descripteurs et schéams de description, prenant en compte des caractéristique de couleur, de forme ou encore de texture.

Quulques travaux de recherche plus récents méritent aussi être mentionnés.

Ainsi, le descripteur GIST introduit dans [OT01] permet d'utiliser conjointement les informations de contour et de texture. Les auteurs proposent d'utiliser des filtres de Gabor (32 filtres) pour obtenir des cartes représentant les structures spatiales moyennes. Chacune de ces cartes est ensuite divisée en 16 sous-régions, qui sont utilisées pour construire un descripteur de dimension 512. Ce descripteur permet de résumer l'information des gradients des différentes zones de l'image, tout en restant relativement compact. Cependant, il est relativement peu robuste face aux transformations géométriques de similarité.

En 2010, plusieurs descripteurs globaux plus robustes aux transformations ont été proposés, tels que le Color and Edge Directivity Descriptor (CEDD) et le Fuzzy Color and Texture Histogram (FCTH) [Cha+10 ; Zag+10]. Ils permettent d'utiliser à la fois les informations de texture et de couleur en utilisant, respectivement, cinq filtres numériques issus de MPEG-7 dans le cas de CEDD, et des ondelettes de Haar pour l'utilisation des bandes de hautes fréquences dans le cas de FCTH. Le descripteur JCD (Joint Composite Descriptor) introduit dans [Zag+10] est également une combinaison des deux descripteurs précédents.

Cependant, ces différentes approches par descripteurs globaux ne sont pas suffisamment robustes en présence de déformations dans les images de requête. Ces descripteurs se révèlent donc peu pertinents pour des applications de réalité augmentée.

A partir du milieu des années 2000, une approche complètement différente a conduit à des avancées significatives dans le domaine de la vision par ordinateur. Il s'agit notamment des méthodes de description locale par points/régions d'intérêt. Ces méthodes sont utiles pour diverses tâches telles que la détection, la reconnaissance et la localisation d'images, d'objets, de visages...

2.2.2 Descripteurs locaux

Les descripteurs locaux sont des vecteurs de caractéristiques qui permettent de décrire localement les aspects visuels d'une image, dans les voisinages d'un ensemble de points d'intérêt détectés en amont. Les points d'intérêt habituellement considérés concernent des régions de l'image de fort gradient selon deux directions, tels que les coins. Leurs capacités à être robustes face aux transformations géométriques tels que la rotation, le changement d'échelle, la translation, etc, les rendent remarquables.

Ces approches impliquent deux étapes distinctes et, la plupart du temps, indépendantes.

La première concerne la détection proprement dite de points d'intérêt. L'objectif primordial ici est la répétabilité de la détection, i.e. la capacité des algorithmes de détecter les mêmes structures de façon invariante par rapport à la position et transformations géométriques, aux conditions d'éclairage ou à la perspective.

La deuxième phase concerne la description proprement dite. Le principe ici consiste à déterminer un vecteur de caractéristiques décrivant l'apparence visuelle du voisinage d'un point d'intérêt.

Chaque image se retrouve donc décrite par un ensemble de points d'intérêt, chacun représenté par un descripteur local. La question qui se pose alors est comment comparer et établir une mesure de similarité entre chaque paire d'images (Figure 2.3).

La solution la plus simple consiste à déterminer des appariements, en comparant de façon exhaustive les descripteurs associés à chaque paire de points d'intérêt possible à l'aide d'une distance native dans l'espace des descripteurs (e.g. distance L2 ou Hamming). Cela permet de déterminer pour chaque point d'intérêt de la première image son plus proche voisin (dans l'espace des descripteurs) dans la seconde. Les deux points sont alors appariés si la distance correspondant est inférieure à un seuil pré-défini.

Une fois l'appariement des points d'intérêt effectué, on peut définir la similarité globale entre les deux images comme une fonction du nombre de points mis en correspondance.

Notons toutefois que le nombre de faux appariements qui apparaît en pratique peut être relativement conséquent, ce qui risque de dénaturer les scores globaux de similarité entre les images. Pour pallier cet inconvénient, on applique souvent des mécanismes supplémentaires de vérification géométrique (qui peut être semi-locale ou globale). Le principe consiste à vérifier la cohérence de tous les points appariés par rapport à une transformation géométrique (e.g., une homographie globale entre les deux images). Cela implique de déterminer conjointement la transformation optimale, les points qui la satisfont ainsi que les appariements aberrants qui doivent être éliminés. Pour cela, plusieurs algorithmes sont disponibles. Parmi les plus populaires, citons RANSAC [FB81] (Random Sample Consensus) ou PROSAC (Progressive Sample Consensus) [CM05].

Dans ce processus, une des étapes clés concerne la recherche des plus proches voisins. La méthode standard, également la plus simple, est la recherche exhaustive (brute force). Néanmoins, les temps de calculs associés deviennent vite prohibitives, notamment lorsqu'il est nécessaire de faire des recherches dans des bases de données comportant des milliers d'images. La solution consiste alors à appliquer des algorithmes de recherche approximative des plus proches voisins (ANN – Approximate Nearest Neighbor) qui puissent offrir une solution plus rapide, même si sous-optimale. Pour cela, il est nécessaire de structurer/organiser les données. Nous pouvons distinguer deux types de structures : arborescentes [FBF77] ou basées sur des fonctions de hachage [IM98].

La première étape de construction des descripteurs locaux est l'étape de détection des points d'intérêt, détaillée par la suite.

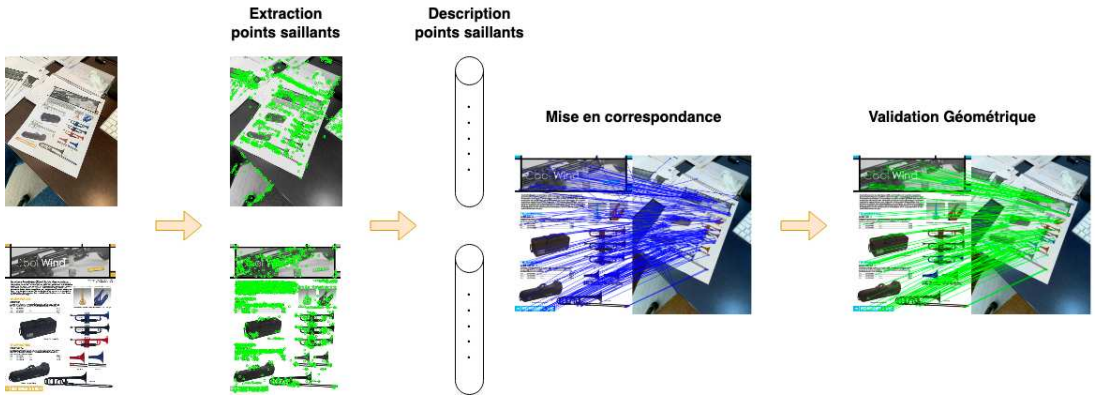


FIGURE 2.3 – Extraction et description de points saillants avec leurs mises en correspondances et une validation géométrique RANSAC

2.2.2.1 Détection de points d'intérêt

Cette étape cruciale permet de limiter l'influence de diverses déformations de l'image et donc d'améliorer la précision et la détection d'images. Un bon point d'intérêt doit normalement être facile à identifier et idéalement rapide à calculer. En outre, il peut porter des informations supplémentaires telles que les coordonnées, mais également la surface de la région d'intérêt, l'échelle de l'image où le point a été identifié et l'orientation dominante des gradients.

Dans la littérature, il y a deux grandes familles de méthodes d'extraction de points clés : les détecteurs de coins et les détecteurs de blobs. Les détecteurs de coins permettent d'identifier les zones avec changements bi-dimensionnels de direction des gradients, ce qui les rend très stables. En revanche, les détecteurs de blobs peuvent identifier des zones avec des propriétés de texture similaires.

2.2.2.1.1 Détections de coins L'un des mécanismes de détection de coins le plus célèbre est le détecteur de Harris [HS+88], qui étend le principe de détection préalablement introduit par Moravec [Mor77]. Les gradients locaux sont estimés via le calcul des dérivées partielles. En construisant la matrice de structure de gradient de second ordre et en évaluant ses valeurs propres, l'estimation des variations locales devient possible, permettant ainsi de discerner si un pixel donné est un coin. Bien que le détecteur de Harris soit capable d'identifier des points d'intérêt de manière invariante à la rotation, il ne supporte pas les modifications d'échelle.

Diverses optimisations ont été mises en œuvre pour surmonter ce dilemme d'échelle. La solution consiste à appliquer une analyse de l'image conduite à multiples échelles, à l'aide d'un opérateur Laplacien. Ce détecteur, dénommé Harris-Laplace [MS01], demeure invariante tant à la rotation qu'au changement d'échelle. Des extensions ultérieures ont également été élaborées pour rendre ce détecteur invariant aux transformations affines [MS04].

La littérature présente également diverses propositions de détecteurs conçus pour une exécution rapide, et donc particulièrement adaptés aux systèmes présentant d'importantes contraintes temporelles et de puissance, tels que les systèmes embarqués ou les applications de réalité augmentée. Le détecteur FAST (Features from Accelerated Segment Test) [RD06] est l'une des principales méthodes rapides de l'état de l'art, qui s'inspire du détecteur SUSAN (Smallest Uni-value Segment Assimilating Nucleus Test) [SB97]. L'idée centrale est de considérer un pixel comme point d'intérêt si son voisinage n'indique pas de similarité en termes d'intensité lumineuse. FAST, quant à lui, considère exclusivement les pixels voisins présents sur le cercle de Bresenham de rayon 3.

Les détecteurs AGAST [Mai+10] et YAPE [LNK18] incarnent également des propositions visant à opérer en temps réel sur des systèmes embarqués. AGAST, une amélioration du détecteur FAST, réduit le temps de calcul tout en préservant des performances équivalentes. YAPE, d'autre part, est salué pour sa simplicité et le minimalisme des opérations requises, ce qui en fait un candidat privilégié pour les applications de réalité augmentée.

2.2.2.1.2 Détection de blobs Les blobs sont définis comme des régions de l'image où l'ensemble des pixels partagent des valeurs similaires. Dans ce cadre, le détecteur Hessian [Bea78] identifie des zones d'intérêt de l'image en se fondant sur la matrice Hessienne, permettant ainsi de découvrir les régions présentant de dérivées de grandes amplitudes selon deux orientations. Similairement au détecteur de Harris, il est uniquement invariant aux rotations. Ce problème est résolu par ses différentes

extensions, telles que le détecteur Hessian-Laplace [MS01], une adaptation multi-échelle, et le détecteur Hessian-Affine [MS05], qui s'appuie sur des stratégies analogues aux détecteurs de Harris afin de rester invariant aux transformations affines.

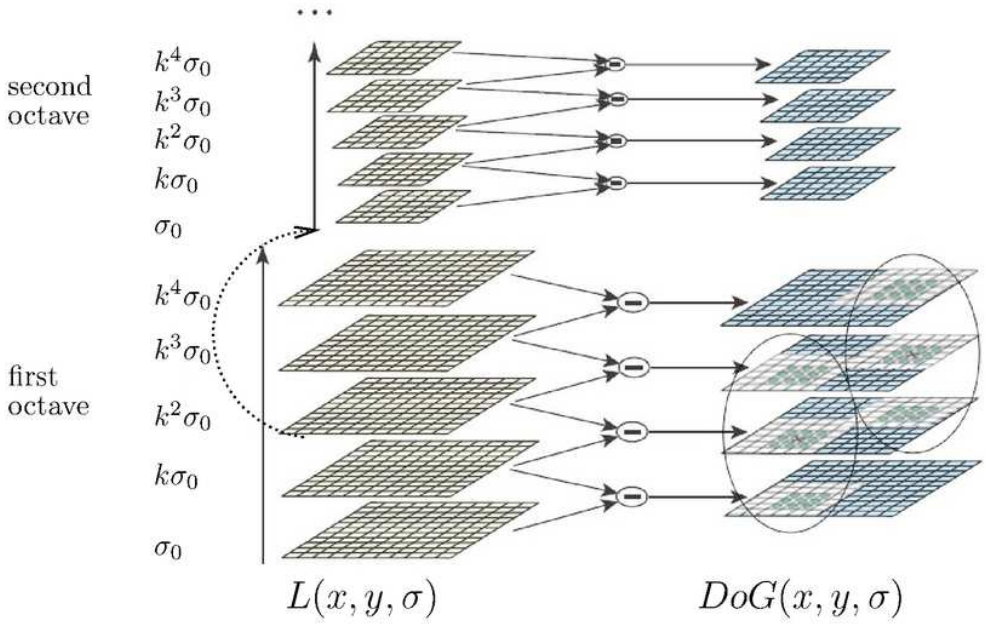


FIGURE 2.4 – Exemple de deux octaves composés de cinq images avec des variations de $K^i\sigma$. Source [Low04]

Cette approche multi-échelle permet d'assurer une invariance aux changements d'échelle en convoluant l'image avec un noyau à différentes échelles. Les détecteurs DoG (Difference of Gaussians) [Low99] et LoG (Laplacian of Gaussian) [Lin98] emploient des noyaux gaussiens. Notons que le DoG, en réalité une approximation de LoG, est utilisé également par le descripteur SIFT et emploie une pyramide à de multiples octaves (Figure 2.4). Les points d'intérêt sont identifiés comme les extrema locaux des différences sur trois niveaux de la pyramide. Néanmoins, ces descripteurs ne sont pas invariants aux transformations affines.

Le détecteur MSER (Maximally Stable Extremal Regions) permet de repérer des régions de l'image uniformes, situées dans un arrière-plan contrasté. Pour identifier ces zones, le détecteur évalue successivement plusieurs seuils. Si certaines régions demeurent stables à travers un éventail de seuils, elles sont catégorisées comme zones d'intérêt. Cette méthode permet d'obtenir des zones d'intérêt invariantes en échelle, rotations, et transformations affines.

Dans le contexte de la reconnaissance de documents, les méthodologies [NKI05; NKI06; INK07; NKI09; NKI07] permettant d'identifier les données textuelles, tels que les mots, en tant que points d'intérêt, méritent d'être mises en évidence. En général, ces méthodes [NKI05; NKI06; INK07; NKI09; NKI07] utilisent le centroïde du mot comme point d'intérêt et peuvent ultérieurement exploiter le voisinage ou la forme pour définir le descripteur.

Ces différentes méthodes, résumées dans le Tableau 2.1 permettent de définir une zone d'intérêt d'une taille adaptée aux caractéristiques locales de l'image et présentant différentes formes d'invariance. La seconde étape à définir un descripteur associés à ces zones, pouvant les caractériser d'une façon

Méthodes	Corner	Blob	Invariant Rotation	Invariant Echelle	Invariant Affine
Harris	x		x		
Harris Laplace	x		x	x	
Harris Affine	x		x	x	x
SUSAN	x		x		
FAST	x		x	(x)	
AGAST	x		x	x	
YAPE	x		x	x	
Hessien		x	x		
Hessien Laplace		x	x	x	
Hessien Affine		x	x	x	x
DoG		x	x	x	
LoG		x	x	x	
MSER		x	x	x	x
Centroides		x	x	x	x

TABLE 2.1 – Résumé de différentes méthodes d'extractions de points clefs

discriminante. Les différents descripteurs de l'état de l'art sont rappelés dans la section suivante.

2.2.2.2 Description de points d'intérêt

Tout comme pour les méthodes de détection, l'état de l'art fait ressortir un nombre important de méthodes de description. Nous pouvons distinguer plusieurs types de descripteurs :

- Descripteurs à base de gradients, s'appuyant principalement sur les orientation des gradients de l'image pour caractériser la texture locale d'une région.
- Descripteurs binaires, qui visent principalement à être le plus compacts et rapides à calculer.
- Descripteurs géométriques, spécialisés notamment dans la reconnaissance de documents, qui cherchent à caractériser les relations spartiales dans le voisinage d'un point d'intérêt.

2.2.2.3 Descripteurs à base de gradients

Proposé en 1999, le descripteur SIFT [Low04] représente une contribution majeure dans ce domaine et reste aujourd'hui une des plus importantes approches de description de points d'intérêt. Par construction, le descripteur SIFT est invariant aux changements de luminosité et aux transformations affines. Pour chaque point d'intérêt, son voisinage est divisé en $n \times n$ cellules. Un histogramme des orientations du gradient est construit pour chaque cellule. Pour cela les orientations sont quantifiées dans un nombre de 8 orientations prototyppes correspondant à un voisinage V8. Dans l'approche traditionnelle, $n = 4$, ce qui donne un total de 16 régions et donc 16 histogrammes de 8 entrées. Le vecteur descripteur est ainsi composé de 128 nombres entiers (Figure 2.5). Pour mettre l'accent sur les informations proches du point d'intérêt, il est possible d'utiliser une pondération par une fonction gaussienne centrée sur le point, dont l'écart-type σ est égal à la moitié de la taille de la fenêtre. Ce descripteur ne prend pas en compte les informations colorimétriques et utilise le détecteur DoG.

Le descripteur SURF (Speeded Up Robust Features) [Bay+08] propose une alternative performante à SIFT tout en réduisant le coûts de calcul associé. De nombreuses similitudes existent entre les deux

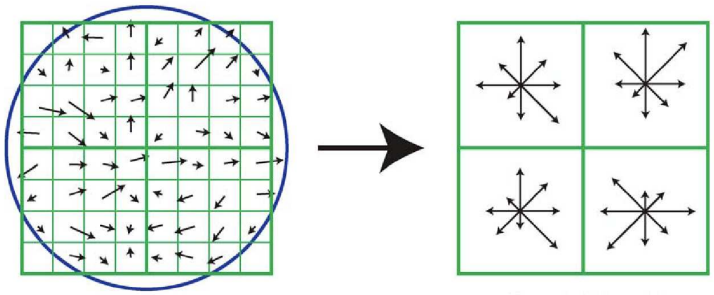


FIGURE 2.5 – Exemple de la construction d'un vecteur SIFT dans le cas d'un fenêtrage $n = 2$. Source [Low04]

descripteurs. Pour la phase de détection, la principale modification réside dans l'approximation du Laplacien de convolution gaussienne pour la détection des extrêmes à différentes échelles, utilisant le détecteur LoG. Concernant la création des vecteurs d'informations, le processus est également similaire à celui des descripteurs SIFT : un voisinage d'études est utilisé en sous-régions de taille $n \times n$. La taille de la sous-région est déterminée en fonction de l'échelle s du détecteur (20 fois s dans l'article d'origine). Cependant, au lieu de construire un histogramme de gradient pour les sous-régions, SURF calcule la somme des ondulations de Haar verticales et horizontales (d_x et d_y). Localement, le vecteur est donc défini comme suit :

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \tag{2.1}$$

Les valeurs absolues $\sum |d_x|$ et $\sum |d_y|$ permettent de fournir la polarité du changement d'intensité. Ainsi, nous obtenons un vecteur de description de dimension $4 \times 4 \times 4 = 64$. Toutefois, il est possible d'augmenter la dimension du descripteur en calculant séparément les sommes de d_x et $\sum |d_x|$ en fonction du signe de d_y et $\sum |d_y|$. Cela permet d'obtenir un descripteur de 128 dimensions tout en conservant la complexité de l'algorithme.

Ces deux descripteurs, SIFT et SURF, sont encore aujourd'hui largement utilisés de nos jours en raison de leur robustesse et de leur complexité acceptable. Ils sont particulièrement adaptés pour des applications ne nécessitant pas de contraintes temporelles importantes.

Toutefois, la complexité de ces méthodes et le nombre d'opérations nécessaire pour la création de tels descripteurs n'est pas compatible avec des applications ayant des contraintes temporelles importantes.

2.2.2.3.1 Descripteurs binaires Comparé aux deux descripteurs susmentionnés, le descripteur ORB (Oriented FAST and Rotated BRIEF) [Rub+11] est quant à lui binaire. L'objectif de ce type de descripteur est d'être plus rapide et efficace afin de fonctionner en temps réel et de nécessiter peu de mémoire. Ces propriétés le rendent d'ailleurs idéal pour les applications de suivi d'images pour la réalité augmentée.

Pour la détection des points d'intérêt, l'algorithme FAST (Features from Accelerated Segment Test) [RD06] est peu coûteux en termes de calcul. Ensuite, pour la partie description, ORB exploite le descripteur BRIEF (Binary Robust Independent Elementary Features) [Cal+10]. Les auteurs partent de l'hypothèse que chaque patch de pixels peut être décrit à partir d'un petit nombre de comparaisons binaire d'intensité après l'application d'un lissage gaussien visant à réduire la sensibilité au bruit. Généralement,

ce vecteur de représentation à une dimension $L = 256$ qui offre un compromis entre performances et efficacité.

Ainsi, grâce à l'association du détecteur FAST et du descripteur BRIEF, ORB permet de construire des descripteurs invariants au contraste, à l'illumination et aux rotations. Cependant, il reste sensible aux variations d'échelle.

Nous pouvons également citer d'autres descripteurs binaires tels que BRISK (Binary Robust Scalable Keypoints) [LCS11] ou FREAK (Fast Retina Keypoints) [AOV12], qui sont invariants à la luminosité et aux transformations affines dans le cas de FREAK.

Ces différents descripteurs binaires sont très largement utilisés et notamment destinés à des applications nécessitant des contraintes temporelles importantes. Cependant, ils sont moins efficaces que les approches SIFT et SURF.

2.2.2.3.2 Descripteurs géométriques Les descripteurs géométriques visent à caractériser l'organisation spatiale des points d'intérêt détéctés. Dans le cas de la reconnaissance de documents, ils sont utilisées pour permettre la construction de vecteurs de description en fonction de l'agencement spatial des mots, sans nécessiter une étape préalable de reconnaissance de caractères.

En premier lieu, mentionnons le descripteur LLAH [NKI05; NKI06; INK07; NKI09; NKI07] avec plusieurs versions plus ou moins complexes et plusieurs expérimentations sur des bases de plusieurs millions de documents [TKI11; TKI12]. Il utilise les centroïdes de chaque mot comme un point clé P, qui peut être obtenu par un détecteur par seuillage. Chaque point clé P utilise un voisinage de n points les plus proches, qui sont organisés dans le sens horaire. À partir de toutes les combinaisons de m points parmi les n points, les caractéristiques peuvent être calculées en fonction de leur agencement. Le descripteur LLAH est notamment robuste par rapport aux distorsions de perspective de la caméra.

L'une des versions les plus simples de LLAH utilise des combinaisons de $m = 3$ points (A, B, C) sur un ensemble de $n = 4$ points, et le rapport de l'aire des triangles $S(A, C, D)$ et $S(A, B, C)$ comme descripteurs.

Plusieurs travaux ont proposé de nouveaux descripteurs [Dan+18] s'inspirant de l'approche LLAH. Un de ces descripteurs, appelé SRIF [Dan+15b], est relativement similaire à LLAH. En effet, le point d'intérêt P est défini en fonction d'un voisinage de n points les plus proches. Ils effectuent ensuite une étude combinatoire de chaque paire possible.

D'autres approches existent et visent à utiliser différentes caractéristiques visuelles, telles que le descripteur "Layout context" [LD07] qui exploite les propriétés des boîtes englobantes d'un mot et de son voisinage, en définissant un nouveau système de coordonnées centré sur le mot. Nous retrouvons également une approche utilisant la forme des mots et leurs fréquences comme caractéristiques [LT08], ainsi que leur longueur [Hul+07].

Nous retrouvons donc plusieurs propositions visant à caractériser les informations textuelles grâce à leurs caractéristiques géométriques et d'agencements. Ces différentes solutions et leurs propriétés sont résumées dans le Tableau 2.2.

2.2.3 Méthodes d'agrégation de descripteurs

Les approches par points d'intérêt conduisent à une représentation d'image de taille variable, représentée comme un ensemble de descripteurs associés aux points/régions d'intérêt détectés. Bien évidemment,

Méthodes	Invariant Rotation	Invariant Echelle	Invariant Luminosite	Invariant Affine
SIFT	+	+	+	+
SURF	+	+	+	+
ORB	+	-	+	-
BRISK	+	+	+	-
FREAK	+	+	+	-
LLAH	+	+	+	-
SRIF	+	+	+	-

TABLE 2.2 – Résumé des différentes méthodes de descriptions de points clefs

chaque image contient un nombre différent de points d'intérêt (et donc de descripteurs), en fonction de son contenu visuel. Pour globaliser la description au niveau de l'image et en même temps assurer une représentation de taille fixe, quel que soit le contenu de l'image, on applique des méthodes d'agrégation.

Le principe de toute méthode d'agrégation repose sur la construction d'un ensemble de descripteurs prototypes, appelés mots visuels, et définissant un dictionnaire visuel. Ces mots visuels sont issus d'un processus de clustering, appliqué dans l'espace des descripteurs à partir d'un ensemble d'images d'entrainement. Dans ce cadre, un des algorithmes de clustering les plus utilisés est le k-means [Mac67; Llo82].

L'approche K-means offre des bons résultats, mais peut être sensible aux données aberrantes. Pour contourner cette problématique, l'algorithme de K-medoids [RK87] propose de ne plus prendre la position moyenne du cluster comme centroïde, mais plutôt utiliser la valeur médiane.

On retrouve l'utilisation de mélanges gaussiennes (GMM : Gaussian Mixture Model) permettant également d'être moins sensibles aux données aberrantes . Cette approche statistique permet de représenter l'espace des caractéristiques comme une somme de gaussiennes avec la moyenne, la variance et l'amplitude de chaque gaussienne comme représentation des clusters k.

Notons qu'il est très important de sélectionner une base d'apprentissage suffisamment variée en termes de contenu visuels (et donc, a priori, de descripteurs associés), permettant de construire un vocabulaire aussi généraliste que possible.

Notons par $F = \{f_1, .., f_m\}$

Les méthodes d'agrégation visent à construire un vecteur global de taille fixe L pour toutes les images (Figure 2.6), qui permet de mesurer la similarité entre deux images en utilisant une simple distance entre ces vecteurs globaux.

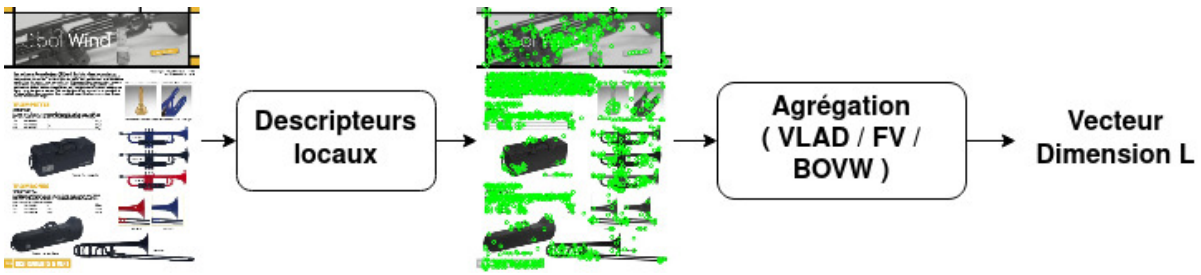


FIGURE 2.6 – Agrégation d'un ensemble de descripteurs locaux en un vecteur compact de dimension L

Une des méthodes d'agrégation de l'état de l'art la plus populaire est celle par Bag of Visual Words.

2.2.3.1 BoVW : Bag of Visual Words

La représentation BoVW ou Bag of Visual Words [Csu+04] s'appuie sur un simple principe de construction d'histogramme, par rapport au dictionnaire de prototypes visuels considérés. C'est une représentation largement utilisée pour effectuer des recherches d'images par similarité, mais aussi pour des objectifs de classification sémantique [OD11].

Chaque descripteur visuel présent dans une image donnée est quantifié à son plus proche prototype (au sens de la distance native dans l'espace des descripteurs) dans le dictionnaire visuel $F = f_0, ..., f_n$, auquel il est dorénavant assimilé. Un histogramme sur le dictionnaire considéré est alors construit, comptabilisant la fréquence relative d'apparition de chaque mot visuel dans l'image.

Le BoVW résultant peut par la suite être utilisé pour représenter de manière unique et robuste l'image considérée. Ce vecteur est utilisé pour comparer les images, trouver des images similaires ou encore classer les images dans des catégories spécifiques [OD11].

L'agrégation par BoVW est simple, rapide et efficace. Néanmoins, comme toute approche par histogramme, elle perd toute information spatiale et structurelle. Elle reste toutefois largement utilisée dans de diverses applications impliquant requêtes par similarité ou catégorisation sémantique.

En ce qui concerne la taille des dictionnaires visuels typiquement utilisés par les modèles BOVW, elle peut varier de quelques milliers à un million de prototypes, en fonction du type d'images et des applications de requête par similarité/catégorisation considérés.

Une deuxième approche d'agrégation, appelée VLAD, est présenté dans le paragraphe suivant.

2.2.3.2 VLAD : Vector of Locally Aggregated Descriptors

L'agrégation VLAD (Vector of Locally Aggregated Descriptors) [Jég+10], [AZ13] permet de construire une représentation vectorielle nécessitant des vocabulaires de plus petites tailles que BoVW, ce qui réduit le besoin en mémoire et la taille du vecteur final. Pour cela, VLAD agrège des descripteurs sur un critère de localité dans l'espace des caractéristiques.

Pour la construction du vecteur VLAD, et comme pour le modèle BOVW, chaque descripteur local x est tout d'abord assimilé à son plus proche prototype $f_i = NN(x)$. La notation NN désigne ici le plus proche voisin (Nearest Neighbor). Ensuite, pour chaque centroïde C_i, on accumule les différences $x - f_i$ des descripteurs x qui lui sont affectés. Cela permet de caractériser la distribution des vecteurs en fonction des centroïdes. Pour déterminer le vecteur VLAD, noté v, il est nécessaire de faire la somme de tous les descripteurs x_j en fonction des centroïdes f_i les plus proches.

Le vecteur VLAD est finalement normalisé (par exemple normalisation L2), comme pour BoVW, pour assurer ses propriétés d'invariance. Plusieurs travaux de recherche ont proposé des modifications/extensions, comme l'utilisation d'une ACP afin de réduire l'impact mémoire [Del+13], ou encore une version hiérarchique de VLAD [ERL14].

Une dernière technique d'agrégation concerne les vecteurs de Fischer, dont le principe est rappelé dans le paragraphe suivant.

2.2.3.3 FV : Fisher vector

L'agrégateur Fisher Vector [PSM10] repose quant à lui sur l'utilisation d'un Modèle de Mélange Gaussien (GMM) préalablement appris sur un ensemble de descripteurs locaux. C'est à partir de ce modèle,

caractérisées par des paramètres tels que la moyenne, la variance et l'amplitude, que l'on peut encoder les descripteurs d'une image. Pour cela, pour chaque descripteur d'une image, on estime la dérivée de log-ressemblance par rapport à chaque composante du modèle. Ces dérivées sont ensuite agrégées afin de constituer le vecteur de représentation de l'image. Cependant, comparée à VLAD, cette approche s'avère plus complexe du fait de l'utilisation d'un modèle de mélanges gaussien et donc plus couteux en termes de calcul.

Les différentes méthodes statistiques évoquées permettent de représenter efficacement des informations visuelles et facilitent l'estimation de similarité entre deux images. Cependant, il est important de noter qu'il n'existe pas dans la littérature d'évaluation et de comparaison exhaustive entre ces méthodes pour la reconnaissance d'images ou de documents provenant de caméras.

2.2.4 Bilan

Dans cette section, nous avons présenté différentes méthodes de construction de descripteurs locaux ou globaux. Ces méthodes sont encore largement utilisées aujourd'hui dans les systèmes de reconnaissance d'images ou de documents.

La littérature comprend également plusieurs études comparatives [Dan+15a] [Dan+16] [Dan+19] sur les performances de différents descripteurs locaux pour la reconnaissance et la localisation de documents à partir de caméras. D'autres travaux visent à évaluer la capacité de ces descripteurs dans le cadre de l'application de la réalité augmentée.

Malgré le bon fonctionnement de ces approches, elles sont de nos jours de plus en plus remplacées par des approches basées sur l'apprentissage profond. En effet, ces dernières ont révolutionné le domaine de la vision par ordinateur en offrant des augmentations spectaculaires de performances dans de nombreux domaines. Présentons ces nouvelles méthodologies dans la section suivante.

2.3 Méthodes par apprentissages

Dans cette section, nous nous intéresserons aux méthodes par apprentissage profond. Pour ce faire, nous définissons tout d'abord les concepts clés, à l'origine des réseaux de neurones, et donnons également quelques éléments d'éclairage sur les aspects d'apprentissage supervisé et non supervisé. Ensuite, nous nous intéressons plus particulièrement aux réseaux de neurones convolutifs, en en exposant principe et fonctionnement. Enfin, nous abordons les méthodes spécifiques à la recherche d'images, à la construction de vecteurs de caractéristiques ou encore à l'estimation de similarité.

2.3.1 L'apprentissage profond

En 1943, deux neuro-scientifiques avaient déjà proposé une représentation d'un neurone théorique inspiré du neurone biologique [MP43]. L'objectif était de proposer un algorithme paramétrable capable de prendre un vecteur de valeurs en entrée, notées $x = [x_1, x_2, ..., x_n]$, et de fournir une réponse y permettant de représenter une classe (Figure 2.7).

Pour ce faire, le neurone pondère chaque entrée avec un poids W_n, puis les agrège en utilisant les additionnant. Cette valeur passe ensuite par une fonction dite d'activation f qui associe l'entrée à une valeur de sortie pouvant représenter une classe. Ainsi, un neurone peut être formalisé comme suit :

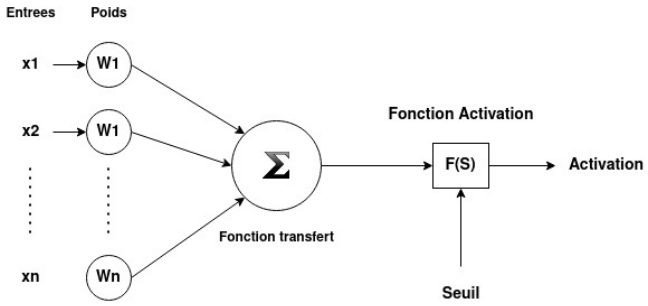


FIGURE 2.7 – Représentation d'un neurone formel

$$y = f(\sum_{i=0}^{L} x_i.w_i) \tag{2.2}$$

Dans le cadre de ce formalisme, ce sont ces poids qui déterminent les résultats. Il est nécessaire donc de les modifier/adapter pour obtenir les résultats attendus, à l'aide d'un processus d'apprentissage conduit sur une base d'entraînement avec vérité terrain.

En 1958, une structure à un seul neurone appelée perceptron [Ros58] a été proposée, offrant une première méthode d'apprentissage à partir de données. Cette structure a permis de réaliser une classification binaire. Cependant, le problème résidait dans la méthode d'apprentissage, qui était trop sensible aux bruits.

Par la suite, l'apprentissage par erreur a été introduit, accompagné d'un changement de la fonction d'activation, qui doit être dérivable, contrairement à la première proposition du perceptron. De nos jours, nous utilisons toujours un algorithme de descente de gradient stochastique pour minimiser l'erreur d'une fonction cible, appelée fonction de perte. Cependant, ces structures à un seul perceptron ne permettent pas de résoudre des problèmes qui ne sont pas linéairement séparables.

C'est en proposant un réseau à plusieurs couches (Figure 2.8), où les opérations simples sont multipliées et connectées entre elles (perceptron multicouches), qu'il devient possible de résoudre des problématiques non linéaires.

Ce type d'approche permet ainsi, à partir d'un ensemble de données, de modéliser un problème afin de tenter de prédire les propriétés de nouvelles données.

Les approches à base de réseaux de neurones se sont fortement démocratisées ces dernières années en raison de leurs performances spectaculaires, notamment dans le domaine de la vision par ordinateur avec l'apparition des réseaux de neurones convolutifs (CNN), décrits dans la section suivante.

2.3.2 Réseaux de neurones convolutifs

En 1998, les travaux de Yann LeCun [LeC+98] ont permis de proposer une première architecture de base pour les réseaux de neurones convolutifs dans le domaine de l'apprentissage profond appliqué à l'image. Les réseaux de neurones convolutifs sont composés de couches successives de différents types (Figure 2.9), comprenant des opérations linéaires et non linéaires. Aujourd'hui, les réseaux de neurones convolutifs sont encore considérés comme la norme pour résoudre diverses taches de vision, telles que la reconnaissance faciale, la détection d'objets, la classification d'images, etc. Ils permettent

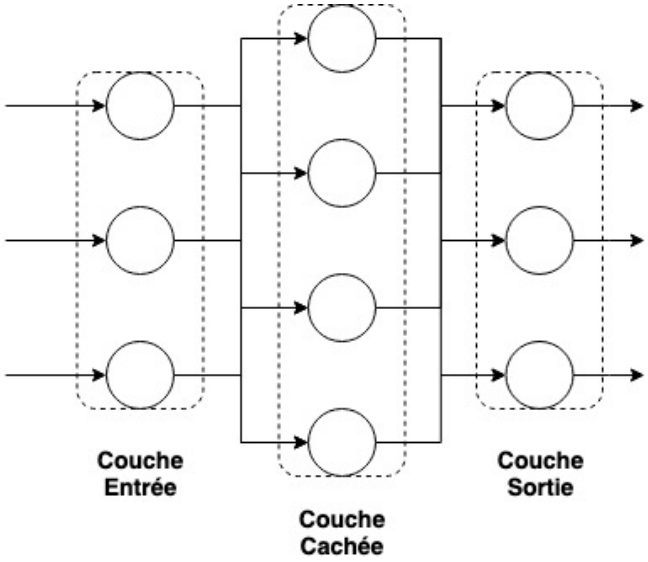


FIGURE 2.8 – Représentation d'un perceptron multicouche

d'extraire des caractéristiques de l'image, allant des plus simples aux plus complexes, selon de multiples niveaux sémantiques.

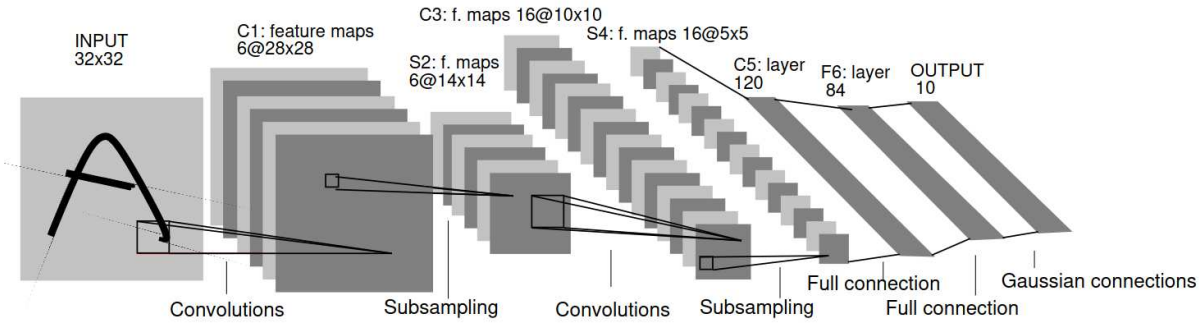


FIGURE 2.9 – Architecture réseaux de neurone convolutif Source [LeC+98]

Ce type de réseaux peut se décomposer en 4 types d'opérations différentes :

2.3.2.1 Couche de convolution

La couche de convolution (Figure 2.10), comme son nom l'indique, opère une opération de convolution entre une image I et un filtre K. Mathématiquement, elle peut être définie de la manière suivante :

$$(I * K)(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) \cdot K(x - i, y - j) \tag{2.3}$$

où $(I * K)(x, y)$ représente la valeur du pixel à la position (x, y) de l'image résultante après l'application du filtre K (Figure 2.10).

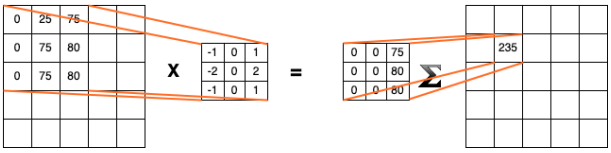


FIGURE 2.10 – Représentation d'un filtre de convolution

Une couche est définie par un ensemble de filtres, paramétrables en termes de nombre, de taille et de pas. C'est lors de la phase d'apprentissage que les valeurs de ces différents filtres sont définis et modifiés afin d'identifier certaines caractéristiques de l'image, plus ou moins complexes.

2.3.2.2 Sous-échantillonnage (pooling)

Les couches de sous-échantillonnage (Figure 2.11) sont généralement utilisées après les couches de convolution. Elles ont pour objectif de réduire la taille des données tout en conservant les informations les plus pertinentes.

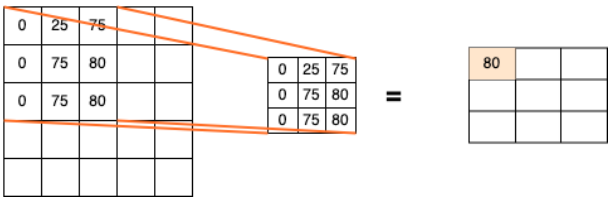


FIGURE 2.11 – Représentation d'un sous échantillonnage

Les opérations les plus couramment utilisées dans les réseaux sont le "max pooling" et le "average pooling", qui agrègent les valeurs en utilisant respectivement la valeur maximale et la moyenne. Cela permet de réduire la taille des données d'entrée et, par conséquent, de limiter le nombre de calculs nécessaires pour les couches suivantes du réseau.

2.3.2.3 Fonction d'activation

Les fonctions d'activation sont des fonctions mathématiques utilisées dans les réseaux à plusieurs reprises. Elles sont souvent placées après chaque couche de convolution et chaque couche entièrement connectée. Ces fonctions permettent d'introduire des transformations non linéaires et d'explorer les données sous d'autres perspectives. Les fonctions les plus couramment utilisées sont illustrées Figure 2.12.

2.3.2.4 Couche entièrement connectée

La couche entièrement connectée, également (fully connected) peut être assimilée à un perceptron multi-couches. Son objectif est de fournir une probabilité d'appartenance à une catégorie ou de fournir une représentation vectorielle des caractéristiques de taille N de l'image d'entrée.

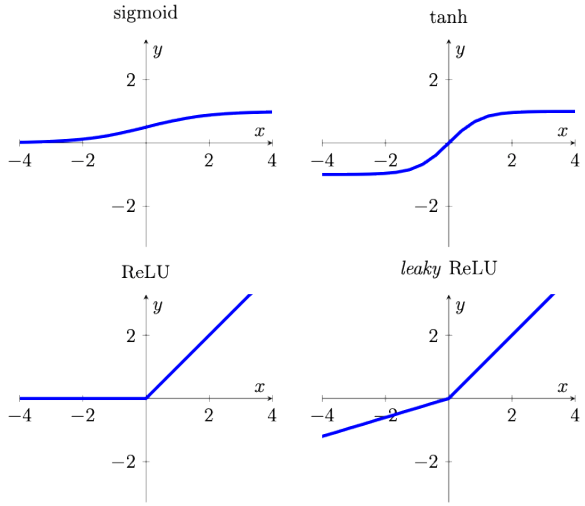


FIGURE 2.12 – Exemples de différentes fonctions d'activations les plus courantes

2.3.3 Les principaux modèles de réseaux de neurones convolutifs

En 2012, l'un des plus importants défis internationaux, appelé ImageNet [Rus+15], a été proposé à la communauté scientifique pour la reconnaissance visuelle à grande échelle. Cette compétition concerne la classification, la détection et la localisation d'images. Elle propose une base de données de 15 millions d'images annotées, réparties en 22 000 catégories différentes. Au fil des années, ce défi a permis de voir l'émergence de modèles proposant, à chaque fois, des améliorations méthodologiques conduisant à des performances de plus en plus élevées.

Le réseau AlexNet [KSH17] a été proposé en 2012 pour répondre à ce défi. Ce modèle est composé de 5 couches de convolutions associées à un sous-échantillonnage de type "max-pooling" et de 3 couches entièrement connectées à l'extrémité du réseau. Il est devenu le premier à obtenir d'aussi bons résultats avec une erreur top-5 de 15,4% en classification. Ces travaux ont également introduit les techniques d'augmentation de données et de régularisation qui sont depuis largement utilisées.

En 2014, le réseau VGG-Net [SZ14], composé de 16 couches dans sa version la plus profonde, a permis d'obtenir une erreur de 7,3% pour le top-5.

De nouvelles propositions d'architecture ont ensuite émergé. Le modèle Inception [Sze+15] est le premier à ne pas utiliser une structure séquentielle des couches pour le traitement des données. Il propose d'intégrer un nouveau module qui effectue des opérations en parallèle, comme illustré Figure 2.13. Ce nouveau module applique d'abord une convolution 1x1 avant les convolutions 3x3, 5x5 et 1x1, puis un max pooling 3x3 pour réduire la dimension des données. Cela permet d'obtenir une extraction d'informations de meilleure qualité et également plus globale, au sein des mêmes couches de convolutions. Le modèle utilise 9 modules Inception, ce qui lui permet d'obtenir un taux d'erreur top-5 de 6,7%. Un autre avantage est qu'il nécessite 10 fois moins de paramètres que AlexNet. Plusieurs améliorations ont ensuite été proposées avec Inception-v3 [Sze+16] et Inception-v4 [Sze+17].

Plus récemment encore, le modèle ResNet [He+16] a été proposé. Il s'agit d'un réseau très profond, pouvant atteindre jusqu'à 152 couches. Le problème des réseaux aussi profonds est lié à la disparition des gradients, qui rend le processus d'apprentissage difficile. Pour pallier cet inconvénient, les auteurs proposent d'intégrer un nouveau module, appelé bloc résiduel, qui permet d'injecter l'entrée

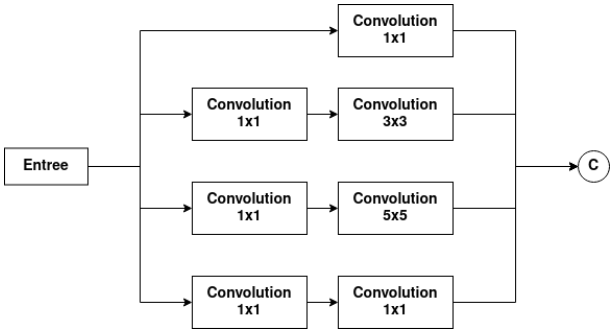


FIGURE 2.13 – Architecture du module Inception

d'un ensemble de couches à la sortie afin de prendre en compte les données passées (Figure 2.14). Ce modèle permet, ainsi, d'obtenir un taux d'erreur top-5 de 3,6%. Remarquablement, le réseau devient plus performant que l'homme (qui présente un taux d'erreur moyen d'environ 5%).

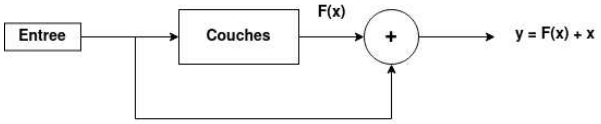


FIGURE 2.14 – Architecture du module résiduel

Enfin, citons également le modèle Xception [Cho17], qui est une extension de Inception. Les auteurs reprennent l'hypothèse des modules Inception en la poussant à l'extrême. Pour cela, ils estiment les corrélations spatiales pour chaque sortie d'une couche, puis quantifient la corrélation entre ces sorties en appliquant une couche de convolution 1x1 (Figure 2.15). Au final, ce modèle permet d'obtenir une meilleure précision que Inception-v3.

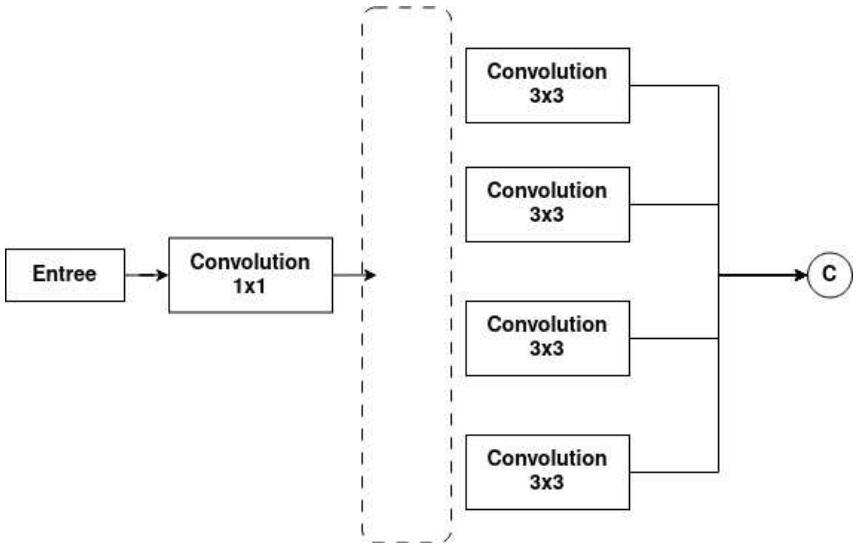


FIGURE 2.15 – Architecture du module résiduel

Cependant, le principal défi de ces différentes approches réside dans la quantité de données requises pour entraîner ces réseaux. De plus, l'entraînement de réseaux très profonds est souvent coûteux

en termes de ressources matérielles (et donc financières). Pour pallier le manque de données, de plus en plus de travaux exploitent des méthodes d'augmentation de données [Won+16; Xu+16] ou de création de données synthétiques [Var+17]. Ces approches permettent d'augmenter artificiellement la quantité de données annotées et de profiter des avancées technologiques des moteurs de rendu 3D. De nombreux travaux s'appuient sur ces modèles existants et parviennent à exploiter leurs connaissances pour les adapter à différentes problématiques. Cette utilisation judicieuse des connaissances préalables offre de nouvelles perspectives pour résoudre des problèmes avec des ensembles de données limitées. Pour cela, ils font appel à l'apprentissage par transfert.

2.3.4 Apprentissage par transfert

L'apprentissage par transfert est une méthode qui permet de tirer parti des connaissances acquises par des modèles neuronaux préalablement entraînés sur d'autre corpus, éventuellement plus larges et plus génériques. Cette approche consiste à utiliser les connaissances et les compétences apprises lors de l'apprentissage d'une tâche pour les appliquer à une autre (Figure 2.16). Au lieu de partir de zéro à chaque nouvelle tâche, nous utilisons des modèles pré-entraînés sur de vastes ensembles de données. Ces modèles ont déjà appris à extraire des caractéristiques significatives et à résoudre des problèmes complexes. En exploitant ces connaissances préalables, le transfert accélère l'apprentissage sur de nouvelles tâches et améliore les performances.

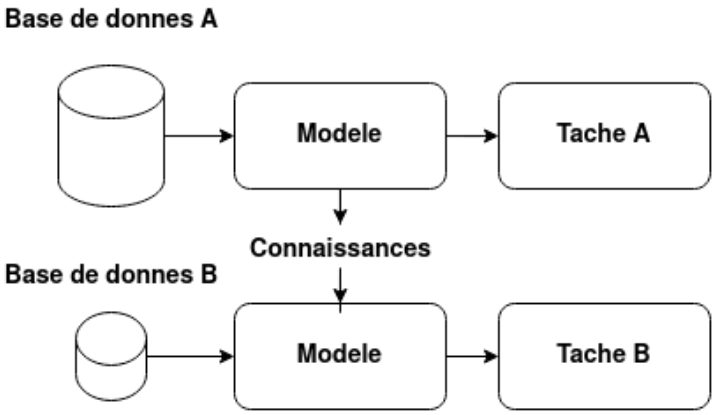


FIGURE 2.16 – Principe de l'apprentissage par transfert de connaissances

Les modèles pré-entraînés servent de point de départ, fournissant des informations initiales aux nouveaux modèles qui sont ensuite affinés et adaptés spécifiquement à la nouvelle tâche. Cette approche est particulièrement utile lorsque les ensembles de données pour la nouvelle tâche sont limités, car elle permet de tirer profit des connaissances déjà existantes pour obtenir de meilleures performances d'apprentissage.

Nous pouvons d'ailleurs distinguer deux types d'apprentissage par transfert : l'apprentissage inductif et l'apprentissage transductif. Dans le cas de l'apprentissage inductif, nous disposons de données annotées à la fois pour les domaines sources et cibles. En revanche, dans le cas de l'apprentissage transductif, nous disposons uniquement de données annotées pour le domaine source.

Dans notre cas, nous nous situons davantage dans le cas de l'apprentissage multi-tâche, c'est-à-dire dans le cadre de l'apprentissage inductif. En effet, l'objectif est d'entraîner à nouveau certaines ou

toutes les couches de certains réseaux à partir de nouvelles données annotées afin de résoudre une nouvelle tâche. Dans la littérature, de nombreuses recherches exploitent cette méthode pour entraîner à nouveau des modèles et concevoir des systèmes de recherche d'images performants.

2.3.5 Les réseaux de neurones convolutifs pour la recherche d'images

À la différence des tâches de classification qui consistent à identifier la classe d'une image donnée, la recherche par similarité s'avère être une tâche plus complexe. En effet, cette tâche peut nécessiter l'identification d'un résultat précis parmi des milliers, voir des millions ou des milliards d'images.

Ces dernières années, de nombreux travaux ont exploité les différents modèles de réseaux de neurones convolutifs présentés précédemment pour créer des vecteurs de représentation, qu'ils soient locaux ou globaux. Afin de tirer parti de ces modèles pré-entraînés, les méthodes d'apprentissage par transfert sont largement utilisées pour mettre à jour les paramètres des différents réseaux, réduisant ainsi le besoin en données d'apprentissage.

Pour concevoir des représentations vectorielles (locales ou globales) d'une image à partir de réseaux de neurones convolutifs, il est possible d'utiliser différentes caractéristiques au sein même du réseau. La première solution consiste à utiliser les informations en sortie des différentes couches de convolution. Cette approche offre l'avantage de conserver les informations structurelles. La seconde possibilité est d'utiliser la sortie des couches entièrement connectées. Cette méthode permet d'obtenir des informations sémantiques de haut niveau, mais elle manque de détails sur l'information structurelle.

Ensuite, ces différentes caractéristiques sont incorporées et agrégées afin de favoriser leur pouvoir de discrimination et d'obtenir des représentations vectorielles globales et/ou locales nécessaires à la recherche et à la reconnaissance d'images. Dans le cas où nous disposons d'un nouvel ensemble de données étiquetées, les méthodes d'apprentissage supervisées peuvent être divisées en deux stratégies distinctes dans le contexte de la recherche par similarité.

2.3.5.1 Apprentissage par classification

L'approche la plus simple et similaire aux modèles de référence présentés précédemment consiste à utiliser une fonction de perte croisée. Ainsi, les descripteurs extraits des régions locales sur les cartes de caractéristiques convolutionnelles peuvent être utilisés directement. En 2018, l'approche DELF (Deep Local Features) [Noh+17] a été proposée, nécessitant un entraînement en deux étapes. La première étape consiste à entraîner l'ensemble du réseau, tandis que la deuxième étape vise à optimiser une couche d'attention spatiale pour la localisation des vecteurs discriminants. Ces différents vecteurs peuvent ensuite être réduits en dimension et utilisés pour la mise en correspondance entre deux images, à la manière de SIFT.

Par la suite, plusieurs fonctions de perte dédiées aux problématiques de recherche par similarité ont été proposées, notamment ArcFace [Den+19]. Cette fonction de perte spécifique a été développée pour améliorer la performance des systèmes de recherche par similarité.

Le modèle DELG [CAS20] (DEep Local and Global features) tire parti de cette fonction de perte pour la conception d'un vecteur de caractéristiques globales. Il est également associé à une fonction de perte croisée et de reconstruction pour les caractéristiques locales. Cette approche permet, contrairement à DELF, d'obtenir à la fois des descripteurs locaux et globaux essentiels pour les moteurs de recherche d'images.

En 2021, l'approche DOLG (Deep Orthogonal Local and Global Features) [Yan+21] a été proposée pour fournir un descripteur global tout en tenant compte des données à la fois au niveau local et global du réseau d'encodage. L'objectif de cette méthode est de simplifier le pipeline de reconnaissance en ne nécessitant qu'une seule étape lors de la recherche de l'image la plus similaire. Au lieu de passer par une étape de recherche des plus proches voisins suivie d'une validation géométrique comme DELG, seule la première étape s'avérera nécessaire.

Pour cela, les auteurs introduisent un nouveau module appelé "Orthogonal Fusion Module", qui prend en entrée les données provenant des branches locales et globales. DOLG permet d'obtenir un vecteur de caractéristiques unique, en exploitant à la fois les informations locales et globales. Comparée à DELF et DELG, cette approche se révèle plus complexe en raison de l'utilisation d'une série de couches de convolution pour la partie locale et des calculs nécessaires pour le module orthogonal. Cependant, les résultats obtenus sur différentes bases de données montrent une supériorité des descripteurs obtenus, malgré leur compacité ($N = 512$).

2.3.5.2 Apprentissage par paires/triplets

Lorsque nous disposons d'une base de données composée de peu d'échantillons, il est possible d'utiliser des méthodes d'optimisation basées sur des métriques de distances. L'objectif ici est de minimiser (maximiser) la distance entre des paires similaires (resp. différentes). Les réseaux siamois, tels que ceux décrits dans les travaux de Gordo et Radenovic [Gor+16; RTC16], tirent partis de ce type d'optimisation, pour apprendre des caractéristiques discriminantes entre différentes classes à partir d'un ensemble de données limitées.

Ces réseaux sont composés de deux réseaux identiques ou plus, partageant les mêmes paramètres. Pendant la phase d'entraînement, la mise à jour de ces paramètres est effectuée de manière identique sur l'ensemble des sous-réseaux. Ainsi, ils permettent de comparer des caractéristiques afin d'estimer la similarité entre des images, ce qui les rend très utiles pour la recherche d'images par vecteurs. Cependant, étant donné que ces réseaux fonctionnent par paire ou plus durant la phase d'apprentissage, ils nécessitent plus de temps pour être entraînés.

Pour l'entraînement de ce type de réseau, nous retrouvons principalement deux fonctions de perte :

2.3.5.2.1 La fonction de perte triple Introduite dans [Wan+14], elle fonctionne à partir de 3 images, une image de référence (ancrage), une image positive et une image négative. L'objectif est de minimiser la distance entre l'ancre et l'image positive puis de maximiser la distance entre l'ancre et l'image négative.

$$\mathcal{L}(A, P, N) = \max\left(0, \|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha\right) \tag{2.4}$$

ou α représente la marge utilisée pour augmenter les distances entre les paires négatives et positives. Les termes $f(A)$, $f(P)$, $f(N)$ représentent respectivement les vecteurs de caractéristiques pour l'ancrage, les images positives et négatives.

2.3.5.2.2 La fonction de perte contrastive Il s'agit d'une fonction couramment utilisée qui reprend le même principe que la fonction précédente. L'objectif cette fois est de minimiser la distance Dw d'une paire positive et de la maximiser pour une paire négative.

$$L_{\text{contrastive}} = (1-Y)\frac{1}{2}(Dw)^2 + (Y)\frac{2}{2}\{max(0, m - Dw)\}^2 \tag{2.5}$$

Des études [Xia+19], [Min+20], ont montré que l'utilisation de fonction de perte triplet et de classification pouvaient considérablement améliorer les capacités d'un réseau. Nous utilisons alors la sortie de ces deux fonctions de perte qui peuvent être ensuite pondérées.

Nous retrouvons différentes propositions de modèles tirant parti de cette architecture pour leurs entraînements tels que le modèle ConVNet [Lee+22]. Ce modèle, divisé en deux parties, permet dans un premier temps d'utiliser d'approches d'agrégation de caractéristiques tels que GeM (Generalized Mean) [TJC20] pour concevoir un descripteur global. La fonction de perte tire parti de l'architecture siamoise et utilise également une fonction de perte dédiée aux problématiques de classification.

La seconde partie de leurs réseaux repose sur un réseau siamois, utilisant des convolutions 4D permettant de comparer les cartes de caractéristiques à plusieurs échelles entre deux images et fournir en sortie une estimation de la corrélation entre les deux images. Cette approche propose également une alternative à l'utilisation de la validation géométrique et des descripteurs locaux.

2.3.6 Les approches par apprentissage pour la détection et la mise en correspondance de points d'interêt

Nous retrouvons dans la littérature des travaux qui s'orientent vers la conception de modèles de réseaux de neurones convolutifs destinés à la détection et la description de points d'intérêt dans une image. Dans cette perspective, SuperPoint [DMR18] a été proposé par le centre de recherche de MagicLeap comme une solution entièrement basée sur l'apprentissage, capable de détecter et de décrire simultanément ces points d'intérêt.

Un des atouts majeurs de SuperPoint (Figure 2.17) réside dans son utilisation d'un unique réseau neuronal convolutif (CNN) pour accomplir ces deux missions. Cette approche intégrée ne se contente pas de produire une carte de caractéristiques illustrant les points d'intérêt potentiels avec pour chaque point détecté un niveau de probabilité, mais elle génère également les descripteurs associés.

L'apprentissage de SuperPoint se distingue par son caractère semi-supervisé. Au départ, il s'appuie sur des données synthétiques qui offrent un accès direct à la vérité terrain pour la détection des coins. Grâce à ces images, le réseau est formé pour reconnaître à la fois les coins et générer leurs descripteurs. Pour renforcer sa performance, SuperPoint est par la suite affiné avec des images réelles, en s'appuyant sur les points d'intérêt et descripteurs précédemment identifiés, ce qui lui permet de se spécialiser et donc faire face a des scénarios plus complexes.

Nous retrouvons naturellement des travaux visant à utiliser ces points d'intérêts et les mettre en correspondance. SuperGlue [Sar+20] est une méthode qui se destine à apparier de manière robuste des descripteurs de points d'intérêt entre deux images. Elle s'appuie sur un mécanisme d'attention pour comparer tous les descripteurs d'une image avec tous les descripteurs de l'autre image simultanément. Au lieu de se limiter à des comparaisons individuelles, elle évalue les relations globales entre les ensembles de descripteurs, ce qui lui permet de déterminer avec précision les correspondances même dans des scénarios complexes.

Le résultat est une matrice de confiance qui donne des scores pour toutes les paires possibles de points d'intérêt entre les deux images. En utilisant cette matrice, SuperGlue sélectionne un ensemble

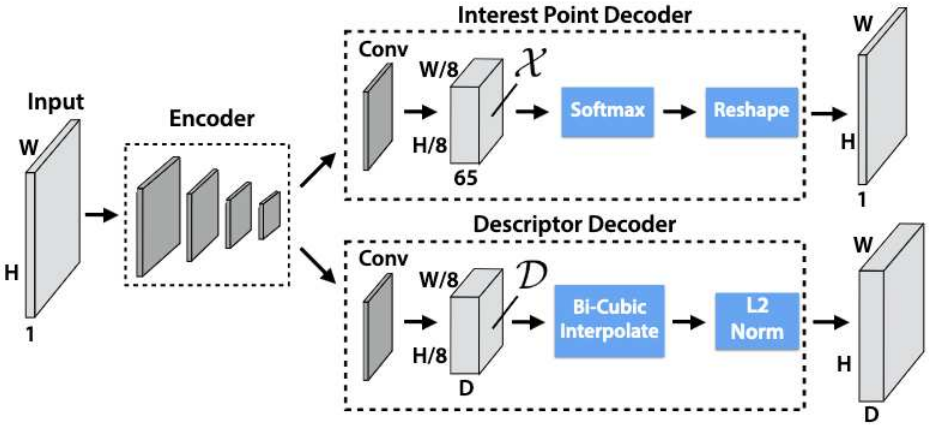


FIGURE 2.17 – Architecture du modèle SuperPoint. Source[DMR18]

optimal de correspondances qui maximise la confiance totale, tout en s'assurant qu'un point d'intérêt n'est associé qu'à un seul autre point. Cette approche globale permet d'obtenir des correspondances plus précises et robustes par rapport aux méthodes traditionnelles.

Plus récemment, le modèle LightGlue [LSP23] a été introduit et qui est un réseau neuronal profond qui apprend à apparier des caractéristiques locales à travers des images. Il se base sur SuperGlue, qui est l'approche la plus performentes pour la mise en correspondance de points d'intérêt. Les différentes optimisations proposées lui permettent d'être plus efficient (que ce soit en termes de mémoire et de complexité de calcul), plus précis et beaucoup plus aisé à entraîner.

2.3.7 Bilan

La littérature fait état d'une riche panoplie de méthodes performantes de reconnaissance d'images. Toutefois, nous observons que les bases de données utilisées, ainsi que les métriques d'évaluation, ne permettent pas de tirer des conclusions définitives quant à leur efficacité dans le contexte de notre étude.

Notre objectif est de développer un système permettant de rechercher des images ou des documents dans une base de données à partir d'une simple photo. Les images de cette base ne proviennent pas de captation terrain contrairement aux images de requêtes. En outre, l'évaluation de notre système s'effectue de façon binaire : le résultat est soit correct, soit incorrect. Le système identifie ainsi soit une image similaire, soit aucune.

2.4 Détection et correction des documents

La littérature regorge de travaux axés sur la détection, l'extraction et la correction de distorsions dans les images ou les documents sur divers supports physiques. Par exemple, les auteurs de [Zha+17] ont mis en place un pipeline complet pour les applications de réalité augmentée, incluant une étape initiale de segmentation d'image. Leurs résultats indiquent que de tels systèmes améliorent significativement la performance des moteurs de reconnaissance d'images ou de documents.

En effet, les systèmes de détection et de segmentation ont l'avantage de réduire les informations non pertinentes dans l'image consultée, concentrant ainsi l'analyse sur les zones discriminantes. On peut classer les méthodes en deux grandes catégories : les méthodes traditionnelles et celles basées sur l'apprentissage.

Pour les méthodes basées sur l'apprentissage, deux principales approches émergent. La première consiste en une détection globale de l'objet cible (ici, un document), approximativement localisé via une boîte englobante. C'est le cas de méthodes largement adoptées comme R-CNN [Gir15] ou YOLO [Red+16]. La seconde approche se concentre sur une localisation plus fine de l'objet, généralement à travers un masque de segmentation. Il existe aussi des architectures capables de fournir ces deux types d'informations simultanément, comme le montrent les travaux de [Ara+18; He+17], mais au prix d'un coût calculatoire plus élevé.

En ce qui concerne les documents, un autre enjeu crucial est la correction des distorsions. Les supports physiques comme le papier sont susceptibles de subir diverses déformations lors de leur manipulation. Cette étape de correction est d'autant plus cruciale pour les documents scannés ou photographiés sous des angles variables, car elle peut influencer la reconnaissance optique de caractères (OCR) et la qualité globale de l'image. Les méthodes de détection et de correction des distorsions sont donc des composantes essentielles pour la reconnaissance automatisée de documents, ce qui s'avère utile dans diverses applications comme la numérisation de documents, la gestion d'archives, ou la reconnaissance de texte et de documents.

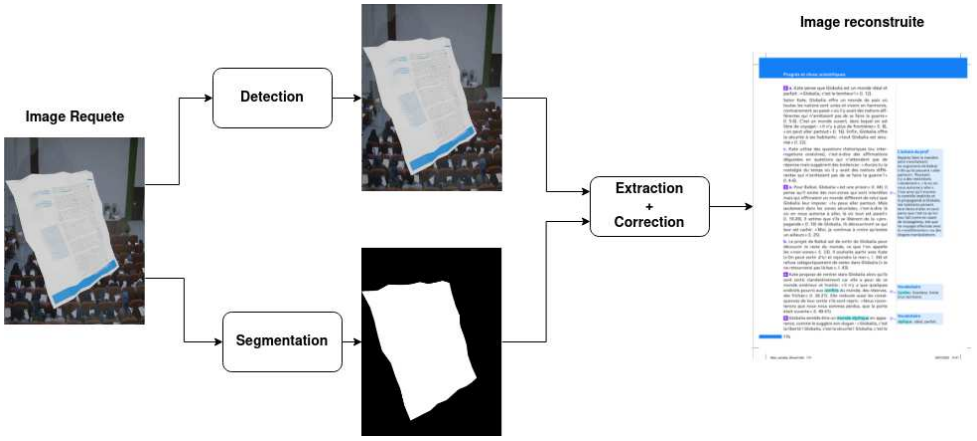


FIGURE 2.18 – Schéma généraliste de détection/segmentation puis d'extraction et correction d'un document pour obtenir une image de qualité

2.4.1 Méthodes traditionnelles de détection et segmentation

La transformation que le domaine de la détection, segmentation et correction de documents a connu au fil des ans est indéniable. Les premières incursions dans ce domaine utilisaient principalement des méthodes dites traditionnelles, sans l'implication d'algorithmes d'apprentissage profond. Avec l'augmentation du nombre d'appareils mobiles dotés de caméras, l'intérêt pour ce sujet a grandi de manière exponentielle.

La mise en place du challenge ICDAR 2015 SmartDoc [Bur+15] a marqué une étape importante en fournissant une base de données qui permet aux chercheurs de travailler sur des problématiques

spécifiques liées à la segmentation de documents, à la correction de perspective, et à la reconnaissance optique de caractères (OCR) sur smartphones.

Avant l'ère de l'apprentissage profond, un certain nombre de techniques étaient utilisées pour détecter et segmenter les documents dans les images. Parmi elles, mentionnons :

1. Le système de détection de lignes LSD (*Line Segment Detector*) [Von+12].
2. L'utilisation de la transformée de Hough pour repérer les formes quadrilatères dans une image, ce qui permet de localiser un document.

Des tentatives ont également été faites pour réaliser la détection en temps réel. Certaines de ces approches [Sko+15; NFG19] exploitent les opérations morphologiques, les cartes de saillances, ou encore combinent l'utilisation de contours et de contrastes d'une image pour la segmentation du document [Tro+20]. Une méthode basée sur l'utilisation de l'algorithme Geodesic Object Proposals a également été testée [KK14; LB16].

Cependant, il est important de noter que les approches basées sur l'apprentissage profond ont pris le dessus dans presque tous les aspects de ce domaine, que ce soit pour la détection, la segmentation ou la correction des documents. Leur capacité à apprendre des représentations complexes des données et à généraliser à partir d'exemples d'entraînement les rend particulièrement efficaces, surtout dans des environnements où l'arrière-plan peut être bruité et où les documents peuvent présenter divers types de déformations.

2.4.2 Les méthodes par apprentissages

De manière analogue aux méthodes utilisées pour la construction de descripteurs, les approches basées sur l'apprentissage profond dominent actuellement le domaine de la segmentation d'images. Parmi ces approches, l'utilisation d'auto-encodeurs est particulièrement répandue [Yan87; HZ93]. Un auto-encodeur est typiquement composé de trois composantes principales, comme illustré Figure 2.19 :

— **Encodeur** : Cette composante est chargée de convertir une image d'entrée en une représentation compacte dans un espace latent.
— **Espace Latent** : Il s'agit du milieu intermédiaire où les données d'entrée sont représentées de manière compressée. Il fait office de lien entre l'encodeur et le décodeur.
— **Décodeur** : Cette dernière étape utilise la représentation compressée issue de l'espace latent pour générer une sortie, qui dans le contexte de la segmentation d'image, peut être un masque de segmentation.

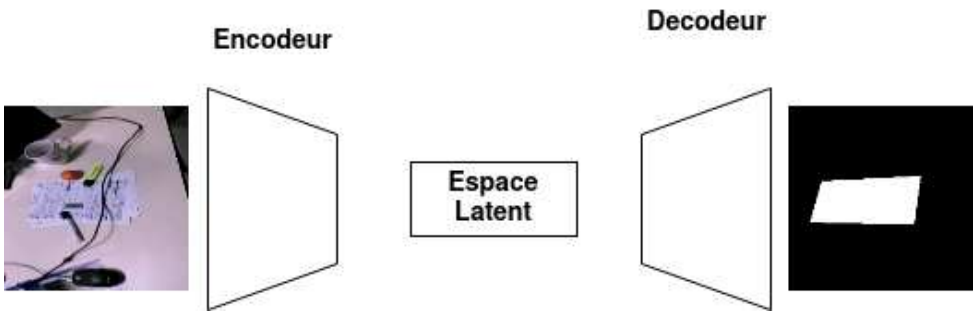


FIGURE 2.19 – Schématisation d'un auto-encodeur pour la segmentation d'image

Cette architecture, notamment l'auto-encodeur, est extrêmement polyvalente et trouve des applications dans divers domaines. Par exemple, elle est utilisée pour débruiter les images. L'espace latent, quant à lui, sert souvent de descripteur d'image en reconnaissance d'images.

La meilleure performance dans le challenge ICDAR a été obtenue par Hu-PageScan [Nev+20], qui utilise une variante simplifiée de l'architecture U-Net [RFB15], essentiellement un autoencodeur entièrement convolutif.

De nombreux travaux ont également exploré l'usage d'auto-encodeurs pour accomplir à la fois la segmentation et la correction de documents. Dans l'approche la plus élémentaire, deux auto-encodeurs en cascade sont employés : le premier se concentre sur la localisation du document, et le second sur sa correction [Ma+18].

Par ailleurs, certaines méthodes exploitent des réseaux neuronaux convolutifs. Par exemple, l'étude présentée dans [JS17] introduit l'usage de ces réseaux pour prédire de manière récursive les coins des documents. Un premier réseau fait une prédiction grossière de la position des quatre coins, et un deuxième réseau, plus petit, affine cette prédiction en boucle. Une autre proposition [Xie+21] prédit la position d'un ensemble de NN points autour du document.

Plus récemment, un pipeline basé sur deux réseaux neuronaux convolutifs a été introduit [Xue+22] pour prédire la grille spatiale d'un document, permettant à la fois son extraction et la correction de distorsions. Cette méthode exploite ensuite une transformée de Fourier pour générer un vecteur de représentation compact, utile pour les recherches en base de données.

2.5 Conclusion

Les moteurs dédiés à la reconnaissance d'images et de documents fondés sur des systèmes de caméra se doivent d'être en mesure de traiter efficacement des images de documents aux contenus hétérogènes, incluant des informations tant graphiques que textuelles. Ils doivent aussi offrir une réponse rapide (inférieure à une seconde) et s'avérer comparables à un système d'identification robuste. Pour atteindre ces objectifs, il est impératif d'exploiter des méthodes de détection pertinentes, des descripteurs locaux et/ou globaux robustes, ainsi qu'un système d'indexation et de validation rapide.

En matière de méthodes de détection, il est essentiel qu'elles soient opérationnelles du côté client afin de minimiser autant que possible la charge processeur sur les serveurs, en particulier lorsqu'elles sont exploitées dans un contexte client. Cette exigence est exacerbée avec l'émergence des applications Web qui, en outre, restreignent les ressources de calcul disponibles.

L'objectif des méthodes de détection peut être double. D'une part, il peut s'agir de guider l'utilisateur en superposant un calque sur l'image probable et ses contours. D'autre part, il s'agit de supprimer les informations d'arrière-plan de l'image de requête, afin de transmettre une image de qualité supérieure au serveur. Toutefois, ces dispositifs ne doivent en aucune circonstance se montrer punitifs, c'est-à-dire affecter négativement la requête d'identification du document.

En ce qui a trait aux descripteurs, qu'ils soient de nature locale ou globale, et qu'ils soient construits par des méthodes descriptives ou par apprentissage, ils incarnent un élément fondamental en devant détecter et décrire non seulement les informations textuelles, mais aussi graphiques. Ils doivent aussi manifester une robustesse pour atténuer l'impact des problématiques soulevées par les images/documents capturés par une caméra. En raison de ces prérequis, les systèmes de localisation d'informations fondés

sur des caméras doivent exploiter des caractéristiques invariantes à la rotation et à l'échelle et/ou résistantes aux transformations affines et de perspective. De plus, afin de contrer les problèmes potentiels engendrés par les images de documents capturées par des caméras, les descripteurs doivent faire preuve de robustesse face à la luminosité, au contraste, au bruit, à l'éclairage, et au flou.

Il a été observé qu'il subsiste un manque de recherches traitant de notre problématique spécifique, exception faite de quelques travaux permettant d'évaluer les descripteurs locaux et certains descripteurs globaux. Effectivement, les bases de données de référence mobilisées pour la comparaison des diverses méthodes proposées ne reflètent pas adéquatement la problématique industrielle.

Avant d'élaborer un nouveau système de reconnaissance pour les images ou les documents, il s'avère nécessaire de procéder à une évaluation des principales méthodes actuelles. Dans une première étape, nous présenterons nos travaux visant à dresser un état des lieux des performances de ces différentes méthodes sur une base de données que nous avons élaborée. Nous présenterons ensuite nos travaux quant à la proposition d'une nouvelle méthodologie pour la reconnaissance d'images.

Chapitre 3

Analyse et évaluations de différentes méthodes pour la recherche d'images

Ce chapitre se penche sur les différentes bases de données actuellement disponibles pour la reconnaissance d'images à partir de caméra, soulignant leurs caractéristiques et leurs limites. Il présente ensuite notre nouvelle base de données conçue pour une évaluation plus précise et robuste de la reconnaissance d'images. Cette base de données est introduite avec une description détaillée de sa constitution, de ses caractéristiques et des avantages qu'elle offre. Le chapitre conclut avec une évaluation de diverses méthodes de reconnaissance d'images, testées spécifiquement sur cette nouvelle base, afin de fournir un aperçu des performances relatives de chaque méthode dans des conditions réalistes.

3.1 Introduction

Dans ce chapitre, nous proposons une évaluation et analyse comparatives des performances des différentes techniques de l'état de l'art pour la recherche d'images au sein d'une base de données. Dans ce cadre, nous nous intéressons plus particulièrement à une des problématiques centrale de nos travaux de thèse, qui est celle de la recherche d'images de documents, acquis à partir d'une caméra grand public dans des conditions incontrôlées.

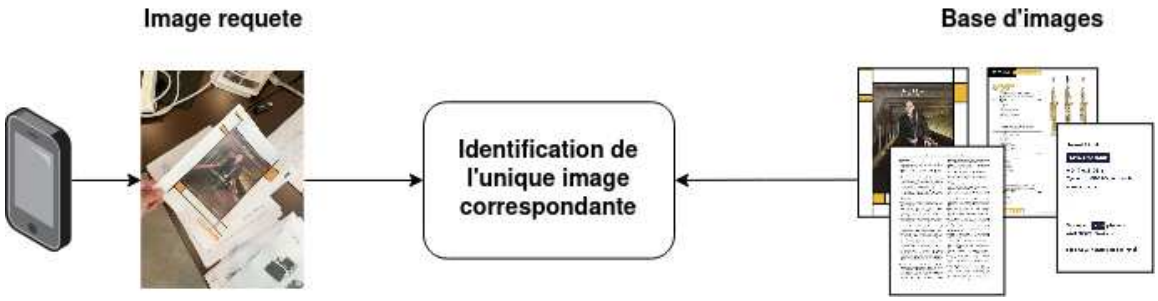


FIGURE 3.1 – Présentation pipeline d'un système de reconnaissance d'image depuis une caméra

De manière plus spécifique, notre objectif est d'évaluer ces techniques dans le cadre précis de l'identification d'une image ou d'un document, capturés via un dispositif caméra, au sein d'un ensemble de données prédéterminées. Selon le contexte d'application envisagée, il convient de déterminer l'unique occurrence d'une image au sein de cette base de données. En l'absence de l'image au sein de l'ensemble, le système se doit d'en notifier l'absence.

Après une présentation rapide des bases de données existantes, pertinentes à notre champ de recherche, nous introduisons une contribution qui concerne la construction d'une base de données spécifique [LPZ22a]. Cette base d'image a été conçue pour refléter fidèlement les types d'images couramment rencontrées dans le secteur industriel, principalement par la société ARGO. Elle vise notamment à illustrer les enjeux liés aux possibles confusions courantes dans ce milieu.

3.2 Bases de données

Pour initier une expérience de réalité augmentée à base de marqueurs naturels, la reconnaissance précise du document concerné est une étape primordiale. Néanmoins, l'état de l'art offre peu de solutions à cette problématique [Zha+17]. Dans ce cadre, le manque de disponibilité de bases de données réalistes et reflétant les difficultés rencontrées en pratique, est un premier défi à surmonter. Pour cela, nous introduisons une base de données dédiée à la reconnaissance d'images de documents [LPZ22a], conçue spécifiquement pour les enjeux associés à la capture par caméra. Elle englobe un large éventail de documents, incluant des catalogues, des factures, des cartes de visite, et se distingue par sa richesse en termes de mises en page, de typographies et de résolutions d'image. De plus, nous proposons un protocole pour la génération de données synthétiques, visant à enrichir tant en volume qu'en variété notre base de données.

Avant de détailler notre approche, analysons les bases de données de la littérature qui puissent être utiles à nos développements.

3.2.1 Bases de données existantes

3.2.1.1 Stanford Mobile Visual Search Dataset

En 2011, Stanford a publié [Cha+11] un ensemble de données contenant des images de produits, de CD, de livres, de repères extérieurs, de cartes de visite, de documents textuels, de peintures et de clips vidéo (figure 3.2). Ces données présentent plusieurs caractéristiques clés : des objets rigides, des conditions d'éclairage très variables, des distorsions de perspective, des encombrements à la fois de premier et d'arrière plans, ainsi que des données de référence proches de la vérité terrain. Les résolutions des images varient en fonction des appareils utilisés pour collecter les images. Au total, la base contient 1200 images de référence et 3300 images de requête. Le point fort de la base de Stanford est sa diversité de contenus, qui est très proche de nos objectifs industriels. Cependant, elle est dépourvue d'images de référence qualifiées de "à risque", c'est-à-dire susceptibles de partager des informations visuelles ou textuelles potentiellement discriminantes, provoquant ainsi des risques de confusion du système de reconnaissance d'image.

FIGURE 3.2 – Exemples d'images provenant de la base de données de Stanford pour chaque categories

3.2.1.2 Bases WikiBook, CartoDialect, Tobacco et données terrains

En 2014, différents travaux portant sur la présentation de nouveaux descripteurs spécialisés pour les éléments textuels [Dan+14; Dan+15a] ont proposé une nouvelle base de données en utilisant des documents de 3 bases de données publiques. La première, WikiBook, contient 700 pages A4 qui ont été converties en format JPEG avec une résolution de 300 dpi. La seconde, CartoDialect, qui comprend des textes en français, est composée de 400 images de grande résolution découpées en plusieurs parties. Enfin, la base Tobacco, est composée de 1291 documents présentant des informations hétérogènes telles que du texte, des logos ou des tableaux.

TABLE 3.1 – Bases de données avec leurs différentes caractéristiques

Noms	Nombres documents	Résolution	Nombres vidéos	Nombres frames
WikiBook	700	2480x3508	1630	24450
CartoDialect	400	9800x11768	2400	36000
Tobacco	1291	1696	3191	47865

Pour la capture des données, les auteurs ont utilisé une caméra (The IPEVO VZ-1 HD) fixée à une distance de seulement 15 cm du document avec une résolution de 1024x768. Le problème de cette base de données provient notamment de la méthodes de capture, qui n'est pas représentative des

problématiques industrielles auxquelles nous sommes confrontés dans le cas d'applications de réalité augmentée.

3.2.1.3 SmartDoc 2015

En 2015, dans le cadre du challenge ICDAR [Bur+15], la base de données SmartDoc a été publiée dans le but de travailler sur les problématiques de détection/segmentation des documents et d'extraction de caractéristiques. Cette base est constituée de six types de documents différents provenant de bases de données publiques et contenant cinq images de documents par classe. Les documents sélectionnés couvrent différents schémas de mise en page et contenus de documents, qu'ils soient entièrement textuels ou qu'ils présentent un contenu graphique élevé.

Pour la capture des données, chaque document a été imprimé à l'aide d'une imprimante laser-jet couleur sur du papier normal au format A4. Les captures ont été ensuite réalisées à l'aide d'une tablette Google Nexus 7. Des vidéos d'environ 10 secondes ont été enregistrées pour les 30 documents dans quatre scénarios de fond différents. Les vidéos ont été enregistrées en résolution Full HD 1920 × 1080 à une fréquence d'images variable. Les 24 000 images provenant des vidéos présentent finalement des distorsions réalistes telles que la mise au point et le flou de mouvement, la perspective, le changement d'éclairage et même des occlusions partielles des pages du document. De plus, la position des documents est également fournie (Figure 3.3).

FIGURE 3.3 – Images provenant de la base de données SmartDoc

Bien que cette base de données présente un volume significatif d'images, elle demeure insuffisante en termes de diversité d'images de référence, se limitant à seulement 30 documents distincts. De surcroît, les documents présentés ne subissent aucune déformation.

En résumé, les bases de données actuelles ne permettent pas d'illustrer l'une des préoccupations prédominantes observées dans l'industrie. En effet, les images présentant des éléments optiquement semblables ou identiques peuvent induire des confusions. Cette question cruciale requiert donc une base de données davantage réaliste pour mettre en évidence ledit enjeu et évaluer les méthodes actuellement utilisées.

3.2.2 Contribution : Base de données naturelles ARGO

Pour remédier aux insuffisances des bases de données actuellement disponibles, nous avons élaboré une nouvelle base de données [LPZ22a], conçue pour mieux refléter les applications de réalité augmentée industrielles ainsi que les systèmes de reconnaissance de documents. Cette base est composée de 596 documents et de 2780 images offrant plusieurs résolutions, allant de 540x720 à 720x1280. Ces images proviennent de quatre dispositifs distincts, de marque et de gammes différents.

Notre collection inclut divers types de documents, tels que des magazines, des catalogues et des articles scientifiques. De plus, nous avons généré des factures, des cartes de visite et conçu des affiches fictives. Ces documents offrent une riche palette d'informations, qu'il s'agisse de contenus textuels (en anglais, français, ou générés aléatoirement en latin) ou de contenus graphiques, et ils sont caractérisés par des mises en page de diverses complexités.

Les captures ont été réalisées de manière à se rapprocher fidèlement des conditions réelles, incorporant diverses scènes, éclairages, potentielles occlusions, résolutions basses, prises partielles et perspectives variées. Notre base de données est structurée en six catégories distinctes de documents, détaillées dans le tableau 3.2.

TABLE 3.2 – Tableau détaillant la base de données que nous avons conçue avec le détail du nombre d'images de référence et d'images de requêtes pour chaque catégories

Catégorie	Images de références	Images de requêtes
Catalogue	100	500
Cartes de visite	79	395
Factures	100	300
Articles	100	500
Affiches	80	400
Modèles d'affiches	137	685

Dans les figures suivantes, nous présentons des échantillons pour chaque catégorie de notre base de données. Ainsi, la Figure 3.4 illustre des spécimens issus de la catégorie "Affiche", tandis que la Figure 3.5 dévoile ceux de la catégorie "Article" et la Figure 3.6 ceux de la catégorie "Catalogue". Ces trois premières catégories démontrent une vaste diversité de contenus. Elles incarnent trois styles distincts de mise en page fréquemment observés dans le secteur industriel.

On retrouve ensuite, dans un second temps, les trois dernières catégories, ayant la particularité de regrouper des images partageant des informations textuelles ou graphiques. La figure 3.7 et 3.8 illustre le cas de factures et de cartes de visites ayant la même mise en page. La figure 3.9 dévoile quant à elle un regroupement d'affiches ayant plusieurs mises en pages similaires.

Au sein de ces dernières catégories, une homogénéité dans la mise en page est donc notable, avec une convergence d'informations textuelles et visuelles. L'intention inhérente à ces classifications est d'accentuer les enjeux liés aux ambiguïtés que les systèmes modernes de recherche documentaire peuvent rencontrer.

Pour illustrer cela, prenons l'exemple de la figure 3.10, concernant la catégorie des modèles d'affiches. Ici, les zones distinctives peuvent présenter des amplitudes variées. De plus, l'information qui les caractérise peut-être de nature textuelle ou iconographique. Cette catégorie se subdivise donc en six sous-catégories distinctes avec divers zones de discriminations (Figure 3.10), comme le stipule le tableau **??**. Au sein de chacune de ces sous-catégories, les zones discriminantes varient en dimensions,

FIGURE 3.4 – Exemples d'images pour la catégorie Affiche

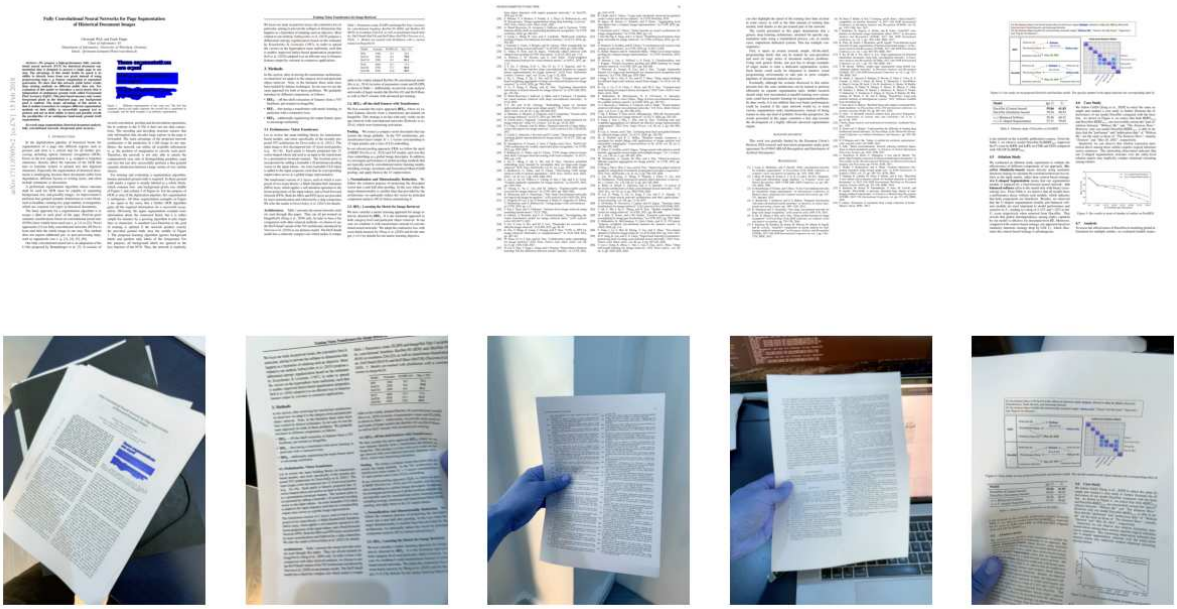


FIGURE 3.5 – Exemples d'images pour la catégorie Article

FIGURE 3.6 – Exemples d'images pour la catégorie Catalogue

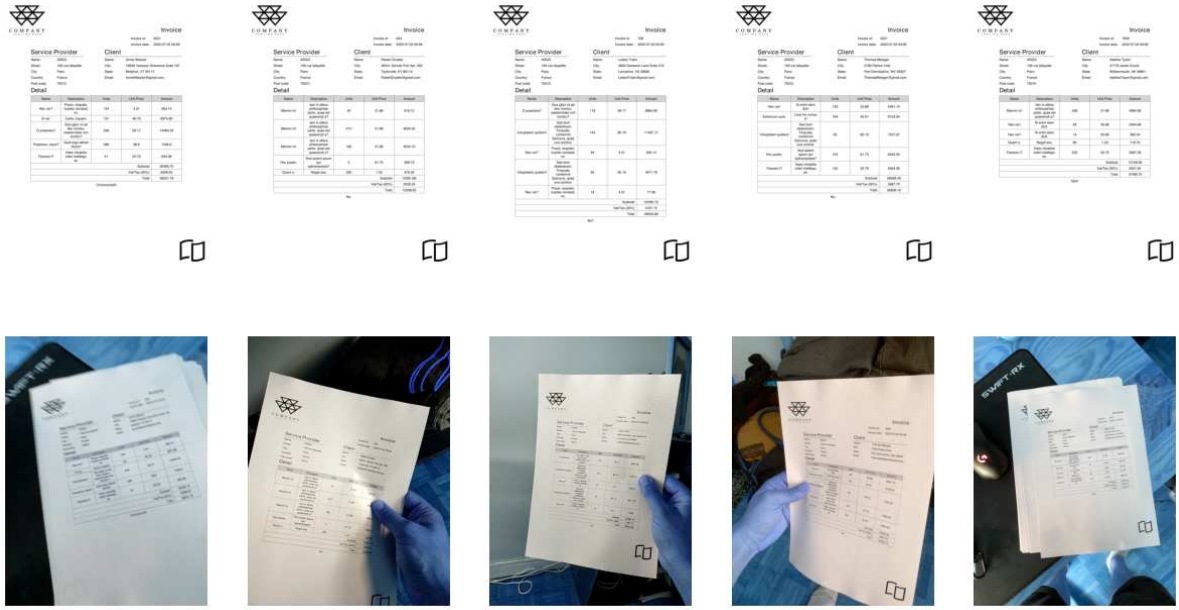


FIGURE 3.7 – Exemples d'images pour la catégorie Facture

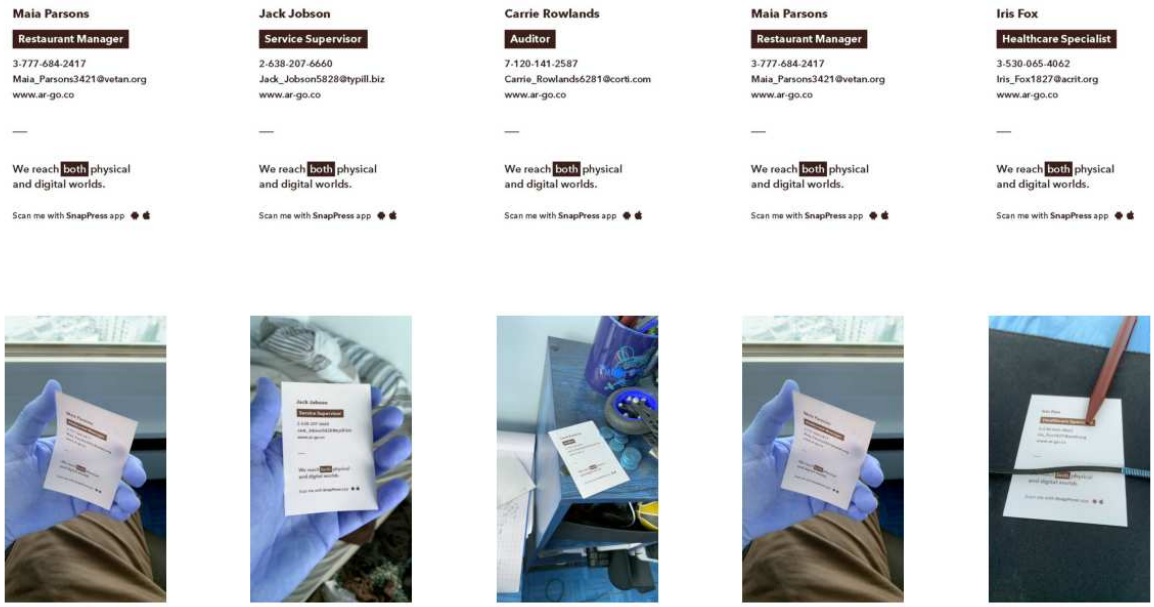


FIGURE 3.8 – Exemples d'images pour la catégorie Carte de visite

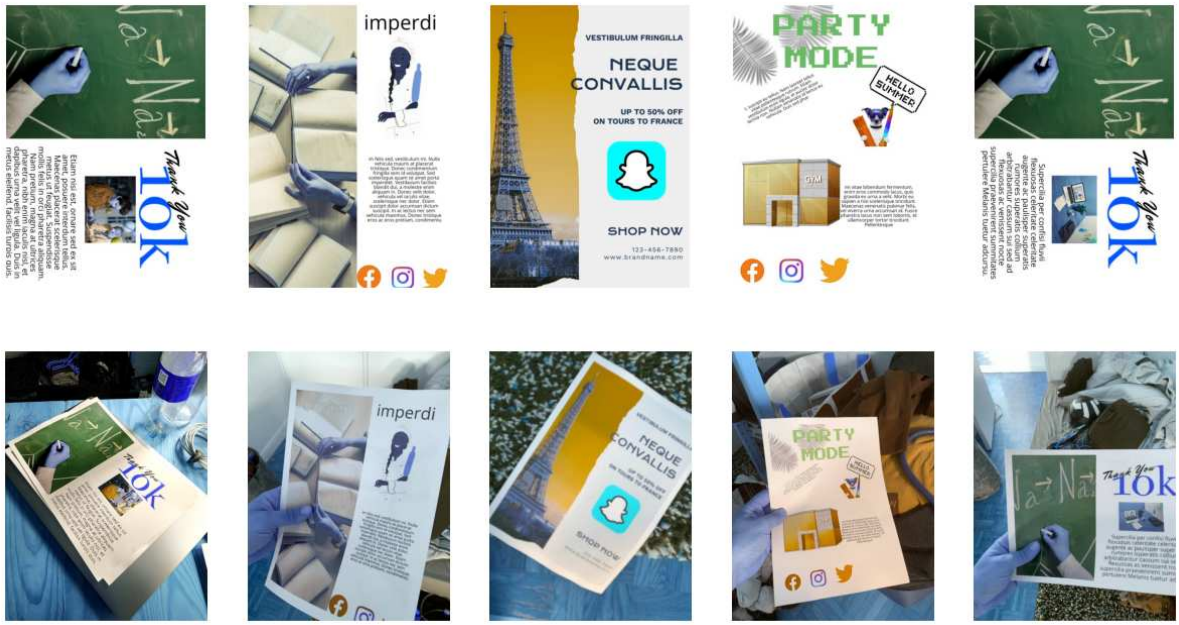


FIGURE 3.9 – Exemples d'images pour la catégorie Modèles d'affiches

influençant par conséquent la complexité du processus d'identification. Il est à noter que l'information permettant la distinction peut être d'ordre visuel ou textuel, conformément à ce que présente le tableau **??**.

(a) (b) (c) (d) (e) (f)

FIGURE 3.10

La base de données ainsi constituée inclut un nombre total de 596 image et 2780 requêtes. Cela est suffisant pour une évaluation expérimentale. Toutefois, dans le cadre des méthodes par apprentissage profond, qui nécessitent des ensembles d'entraînement plus conséquent, il est nécessaire de considérer des bases de dimensionnalité beaucoup plus importante. Pour répondre à ces enjeux, nous avons élaboré une base de données synthétique, décrite dans la section suivante.

3.3 Base de données synthétiques

Les systèmes de réseaux de neurones tirent leur efficacité grâce à leur capacité de généralisation. Toutefois, ils nécessitent un grand nombre de données pour éviter les phénomènes de sur-apprentissage (*overfitting*).

Comme notre base de données naturelle ARGO de ne contient pas assez d'image de terrain et de référence, nous avons décidé d'utiliser des données synthétiques pour l'apprentissage. Notons qu'il s'agit d'une pratique courante dans les systèmes récents d'apprentissage en profondeur [Shr+17; Var+17]. L'avantage est de pouvoir contrôler les variations de données telles que les distorsions, perspectives, géométriques, l'éclairage...

Notre système de construction de données synthétiques a donc pour objectif de créer, pour une image donnée, un ensemble d'images pouvant se rapprocher d'une vérité terrain.

Afin de rendre les images pratiques pour des objectifs d'apprentissage, elles sont normalisées à une taille fixe de 512x512 pixels. Cela permet d'éviter les problématiques de changement de ratio entre l'image de référence et les images de "terrains" pour l'étape d'apprentissage. En ce qui concerne le contenu, les images utilisées seront un ensemble de documents que nous avons récupérées sur la base de données de production de la société ARGO.

3.3.1 Déformation et projection des images

La première étape de notre moteur de génération de données est de simuler des déformations du document. Lors de la création de ces distorsions, nous suivons plusieurs directives empiriques :

Noms	Nombres documents	Nombres photos	Taille surface discriminante	Type informations discriminantes
Mise en page 1	29	145	32%	Texte
Mise en page 2	30	150	36%	Texte/Figure
Mise en page 3	28	140	31%	Texte/Figure
Mise en page 4	20	100	57%	Figure
Mise en page 5	20	100	44%	Texte/Figure
Mise en page 6	10	50	2%	Texte

TABLE 3.3 – Tableau détaillant la répartition des différents mise en page dans la catégorie modèle d'affiches avec la surface et le type des informations discriminantes

FIGURE 3.11 – Exemples de documents provenant de la base de données de la société ARGO qui sont des images actuellement dans la base de données de production

— Un document papier est un objet localement rigide. Il ne se dilate, ni ne comprime. La déformation en un point se propage dans son voisinage.
— Il existe deux types de déformations : des plis et des courbes se propageant comme ondulations du papier. En pratique, il existe généralement un mélange de ces deux distorsions de base.

Pour générer de telles déformations, nous procédons en deux étapes :

3.3.1.1 Déformation de l'image par maillage

Dans un premier temps, pour une image donnée I, nous générons un maillage nommé M constitué de $N \times N$ points espacés équitablement entre eux permettant de modifier la position de ces points.

Nous sélectionnons un point p_n de façon aléatoire dans le maillage M comme point de déformation initial. Une déformation, représentée par un vecteur notée v et associée au point courant est également générée de manière aléatoire. La propagation de v sur les autres points p_i de M est calculée comme $p_i + wv$, avec w comme poids pour chaque point. le poids w_i peut être défini par deux méthodes suivants le type de déformation souhaitée :

— Pour les plis : $w_i = \frac{\alpha}{d_i + \alpha}$
— Pour les courbes : $w_i = 1 + d_i^{\alpha}$

avec d_i étant la distance normalisée d entre chaque point pi de M et p_n. La valeur α permet de jouer sur la puissance de propagation de la déformation v. Plus la valeur α est grande, plus la déformation associée w est importante. A contrario, une valeur plus petite du paramètre α conduira à des déformations plus localisées. À partir de ce maillage, nous pouvons alors déformer l'image d'origine au niveau de chaque pixel via une interpolation linéaire (Figure 3.12).

FIGURE 3.12 – Exemple d'une image déformée après modification du maillage

3.3.1.2 Génération de transformations perspectives

La seconde étape du moteur de génération de données est de créer de manière aléatoire des projections perspectives, représentées par des matrices d'homographies. L'objectif de ce type transformation est de produire des angles de vues crédible et proche d'une vérité terrain.

Ce type de matrice 3×3 est utilisé en géométrie projective afin de représenter la transformation entre deux plans projectifs. Elle permet de transformer les coordonnées $[u, v]$ d'un point dans un plan en des coordonnées $[u', v']$ dans un autre plan projectif, comme décrit dans l'équation suivante :

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \sim \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & 1.0 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Dans le but de simplifier la génération de matrice d'homographies aléatoires pour la création d'images synthétiques, nous avons décidé de générer des matrices d'homographie $\overline{H_n}$ normalisées comme suit :

$$\overline{M_n} = M H_n M^{-1} \tag{3.1}$$

avec

$$M = \begin{bmatrix} w & 0 & -1 \\ 0 & h & -1 \\ 0 & 0 & 1.0 \end{bmatrix} \tag{3.2}$$

où M_n est la matrice d'homographie non normalisée. Nous pouvons donc ensuite facilement générer des matrices avec des valeurs H_{ij} comprises entre -1 et 1. De cette manière, chaque matrice d'homographie peut être appliquée à n'importe quelle image, quelle que soit sa résolution. Quelques exemples sont présentés figure 3.13.

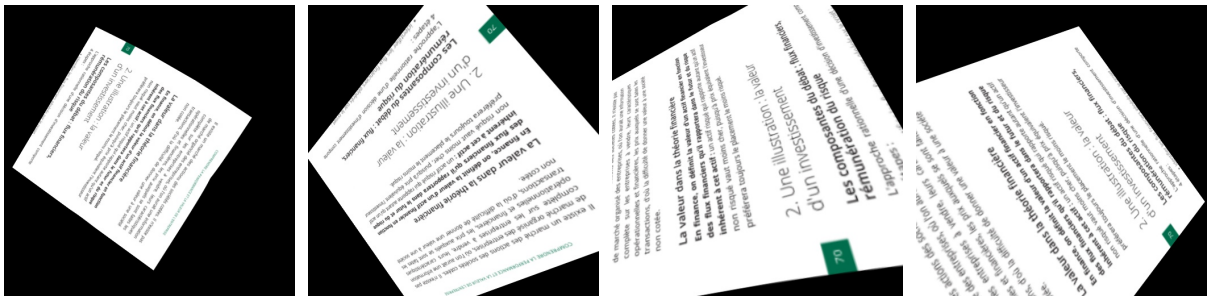


FIGURE 3.13 – Exemple d'une image avec différentes projections perspectives

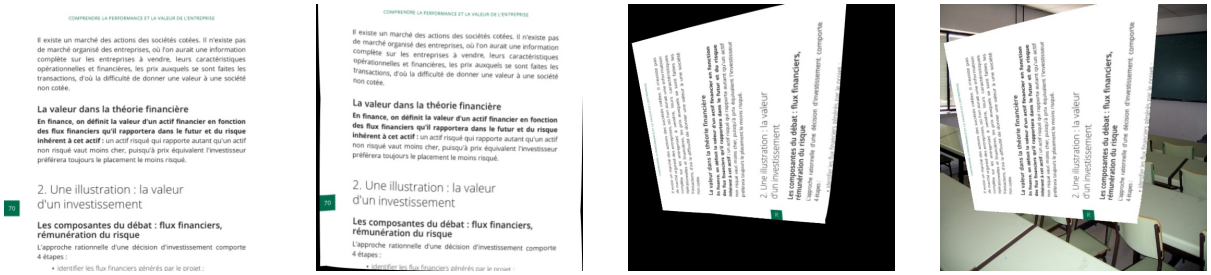
3.3.1.3 Système complet de génération de données

C'est à partir de ces deux étapes, i.e. perturbation du maillage et génération d'homographie, que nous pouvons mettre en place un pipeline permettant de construire des ensembles d'images synthétiques se rapprochant d'une vérité terrain. Comme illustrée dans la figure 3.14, la génération de données suit 3 étapes :

— Génération du maillage et interpolation pour déformation de l'image
— Génération d'une matrice H pour projection
— Ajout d'un arrière-plan à l'image aléatoire

3.3.2 Génération de données synthétiques avec Blender

Nous avons également développé une version de notre système de génération de données synthétiques avec Blender [Com18] (Figure 3.15). Blender est un des logiciels d'animation et de modélisation 3D



(a) Image de référence (b) Image déformée (c) Image projetée (d) Ajout arrière-plan

FIGURE 3.14 – Visualisation de l'évolution d'une image synthétique produite

open-source les plus populaires qui offre une polyvalence exceptionnelle pour la création de données synthétiques. L'un des avantages clés de Blender réside dans sa prise en charge native des scripts Python. Cela permet aux utilisateurs d'automatiser la création de scènes 3D, en générant rapidement et efficacement une variété de configurations sans nécessiter d'interaction manuelle intensive. Par exemple, un script pourrait définir des positions spécifiques pour les objets, définir des paramètres de caméra ou même animer des éléments au sein de la scène.

En outre, Blender facilite la configuration de composants de scène essentiels telles que la caméra et les sources lumineuses. En utilisant des scripts Python, il est possible de définir précisément les angles de vue, la distance focale, l'intensité lumineuse, la position des sources de lumière, et bien d'autres paramètres. Cette automatisation assure une uniformité à travers les scènes tout en permettant une grande variabilité lorsque nécessaire, offrant ainsi une flexibilité optimale pour la création de données synthétiques adaptées à diverses applications.

Un autre atout majeur de Blender est sa capacité à charger des images planes et à les déformer (Figure 3.16). Il est possible d'importer une image plane, de l'appliquer à un maillage et ensuite de déformer ce maillage selon des besoins spécifiques. Ce processus conserve les informations du maillage, ce qui est essentiel pour analyser la manière dont les déformations affectent l'image. Cette fonctionnalité est particulièrement utile pour simuler des scénarios réels où des objets plats, comme des affiches ou des peintures, pourraient être soumis à des déformations dues à des facteurs environnementaux ou à des interactions physiques.

Finalement, pour la génération de notre base synthétique, nous avons regroupé 3 000 images de documents afin d'avoir la plus grande diversité possible de mise en page, d'éléments graphiques et de textes. Pour chaque document généré, nous avons produit 30 images synthétiques et enregistrées pour chaque image générée des informations telles que le masque, la matrice d'homographie et l'image déformée. Nous avons finalement une base de données d'apprentissage de 90 000 images.

3.4 Méthodes retenues et métriques d'évaluation

Introduisons en premier lieu les différentes méthodes retenues pour notre évaluation comparative.

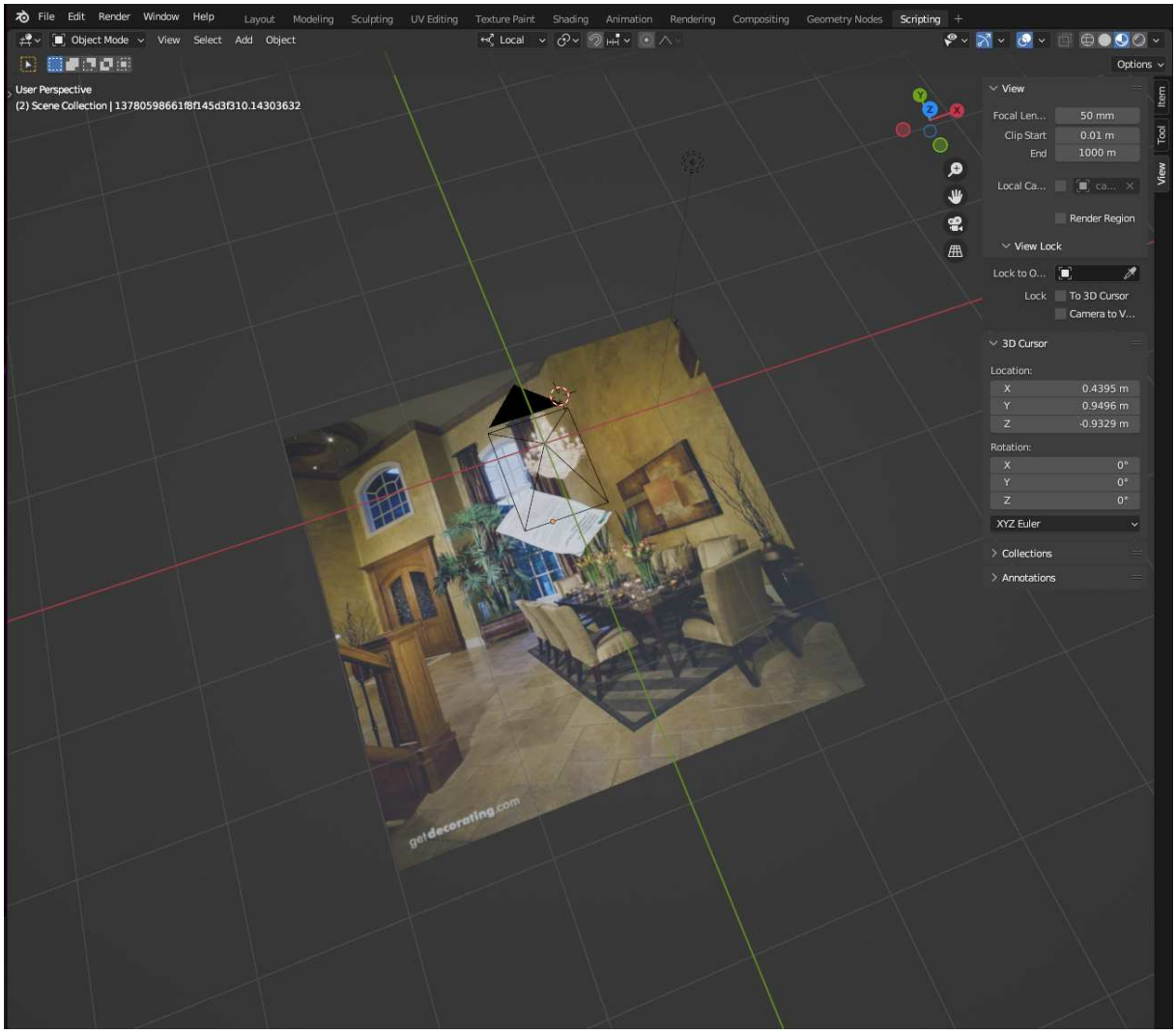


FIGURE 3.15 – Vue du logiciel Blender lors d'une génération automatique de scene

3.4.1 Les différents pipelines

Comme mentionné précédemment, nous distinguons deux types d'approches pour la recherche d'images : les méthodes basées sur la description et celles basées sur l'apprentissage. Néanmoins, quelle que soit l'approche, les processus de recherche présentent des similitudes. Deux étapes fondamentales ont été identifiées, comme nous l'avons présenté dans le précédent chapitre.

La phase "hors-ligne" concerne la construction des vecteurs descriptifs pour toutes les images de la base de données. Elle englobe aussi la mise en place de la structure de recherche par approximation. En raison de l'expansion de la taille des bases de données, spécifiquement des images, les méthodes de recherche des plus proches voisins, ont gagné en importance, devenant un domaine de recherche à la fois populaire et crucial. Ces méthodes occupent une place prépondérante dans de nombreuses tâches et applications, telles que la recherche d'information, la reconnaissance de motifs [DHS73], d'images, de vidéos [Fli+95] ou de documents.

L'objectif principal est de fournir rapidement une liste de candidats potentiels sans nécessiter l'explo-

FIGURE 3.16 – Exemples d'images synthétiques depuis Blender

ration intégrale de la base de données. Nous pouvons définir la recherche du plus proche voisin comme un ensemble de n points $P = p_1, ..., p_n$ dans un espace de dimensions M, avec une requête $q \in M$ pour laquelle nous souhaitons trouver l'élément $NN(q, P) \in P$ le plus proche de q. Pour cela, nous utilisons une distance d qui satisfait $NN(q, P) = \arg\min_x d(q, x) \forall x \in P$.

La méthode standard, également la plus simple, est la recherche exhaustive (brute force) qui se comporte de manière linéaire. Cela signifie que le temps de calcul augmente avec la taille de la base de données, mais garantit la recherche du plus proche voisin, car toutes les combinaisons sont testées. Ce type de méthode est donc inenvisageable dans notre cas (base de données pouvant être composée de plusieurs milliers de documents).

C'est ainsi que les algorithmes de recherche approximative des plus proches voisins (ANN) ont été développés dans le but d'être rapides et de conserver le même niveau de précision que la recherche exhaustive. Pour cela, il est nécessaire de structurer/organiser les données. Nous pouvons distinguer deux types de structures : arborescentes [FBF77] ou basées sur des fonctions de hachage [IM98], de manière que l'opération $NN(q, P) \in P$ soit rapide et efficace.

Pour l'ensemble des pipelines que nous avons conçus, nous avons choisi d'utiliser la bibliothèque Faiss [JDJ19]. Elle offre une prise en main aisée des méthodes arborescentes ou basées sur des fonctions de hachage. En outre, cette bibliothèque facilite la mise en place d'un système de production et est particulièrement efficace, permettant notamment l'ajout ou la suppression de vecteurs sans reconstruire intégralement la structure.

La phase "en ligne", quant à elle, englobe les différents processus de recherche et de validation nécessaires pour identifier l'image cible à partir d'une image d'entrée. Selon les méthodes employées, les processus de validation peuvent varier.

3.4.1.1 Méthodes par description

Dans le cadre des méthodes descriptives, nous avons développé un pipeline modulable, représenté Figure 3.17, spécialement optimisé pour la recherche d'images au sein d'une base de données. Ce système permet une fusion souple des descripteurs locaux, tels que SIFT ou SURF, avec différentes méthodes d'agrégation, notamment Fisher Vectors, VLAD ou encore Bag of Visual Words. Pour évaluer les performances de ces différentes combinaisons, nous avons suivi le protocole et les paramètres décrits ci-après.

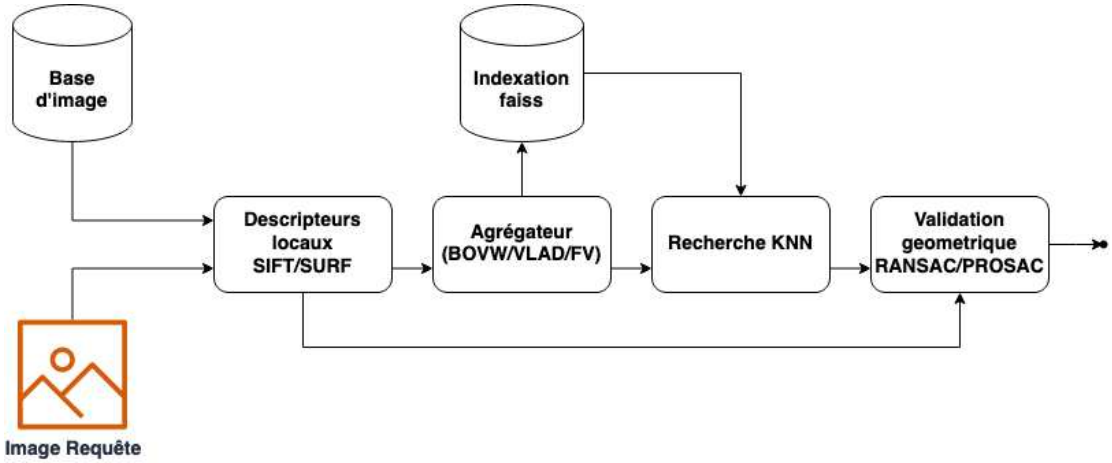


FIGURE 3.17 – Pipeline de reconnaissance d'images intégrant les options des descripteurs SIFT ou SURF, ainsi que les agrégations BOVW, VLAD, ou FV

Toutes les images, qu'elles servent de référence ou de requête, seront normalisées à des dimensions prédéfinies : 640 pixels pour les images de référence et 720 pixels pour les images de requête. Par la suite, nous limiterons le nombre de descripteurs locaux par image à 600.

Concernant les systèmes d'agrégation, l'objectif est de trouver une valeur optimale du nombre de clusters afin de trouver un compromis entre performance et impact mémoire. En effet, nous avons effectué une évaluation rapide de la combinaison VLAD + SIFT sur 16, 32, 64 et 128 clusters sur notre base de données.

Les résultats présentés dans le tableau 3.4 démontrent une stagnation de la précision malgré l'augmentation du nombre de clusters. En effet, en passant de 64 à 128, nous n'obtenons qu'un gain de 0.5 sur la précision $K - NN$ mais en doublant l'impact mémoire du fait de la taille des vecteurs descripteur. Nous opterons donc pour une dimension fixé à $L = 4096$ qui semble un bon compromis entre performance et impact mémoire.

Nombre Clusters	16	32	64	128	256
Dimension Vecteur	2048	4096	8192	16384	32768
$P@40$	0.59	0.73	0.81	0.86	0.86

TABLE 3.4 – Exemple de dimension et impact mémoire de VLAD + SIFT en fonction du nombre de clusters

Durant la phase de recherche et de validation, nous appliquerons une stratégie de recherche accélérée, ne considérant que les 40 candidats les plus pertinents. Chacun de ces candidats sera comparé individuellement à l'image de requête en utilisant les descripteurs locaux et une validation géométrique. Le résultat le plus adapté sera choisi selon le nombre maximal de correspondances. Cependant, des seuils seront définis pour valider ces résultats, en particulier en ce qui concerne le nombre minimal de correspondances validées et le ratio entre les correspondances possibles et validées, établissant ainsi la validation finale du résultat. Nous utiliserons également un seuil afin de vérifier qu'il n'existe pas une ambiguïté entre le meilleur résultat et le second.

Nous avons fait le choix d'utiliser une recherche $40 - NN$ car elle permet d'obtenir des temps de

response relativement restreint. De plus, nous avons fait le choix de ne pas augmenter afin de mettre en evidence les problematiques liees a la precision $P@K$.

3.4.1.2 Méthodes par apprentissage

Dans le contexte des méthodes orientées par apprentissage profond, nous avons évalué trois approches : DOLG, ConvNet et GeM (Figure 3.18). Notre objectif était d'explorer des catégories de méthodes ayant des procédures de validation différentes.

Pour DOLG, la phase de recherche approximative est suffisante pour produire une réponse. Cette méthode génère un vecteur singulier qui intègre les informations locales et globales. La recherche est donc effectuée directement au sein de la base de données. Cependant, il est possible d'introduire des critères de validation basés sur la distance, permettant de ne prendre en compte uniquement les résultats avec une faible divergence. Comme pour les méthodes d'agrégation des descripteurs locaux, un critère de pertinence peut être établi pour juger de la validité d'un résultat et réduire les faux positifs lors d'une requête.

Dans le cas de GeM, nous avons introduit une étape de reclassement comme pour les méthodes par description, mais utilisant SuperPoint [DMR18] pour l'extraction de points d'intérêt et une mise en correspondance et une validation en utilisant LightGlue.

ConvNet fonctionne également suivant un processus en deux phases. La première étape implique la sélection d'une liste initiale de k candidats comme Dolg et GeM. Cependant, en ce qui concerne la validation, ConvNet utilise deux images en entrée pour évaluer le degré de corrélation entre elles, sans avoir recours à une vérification géométrique telle que PROSAC ou RANSAC.

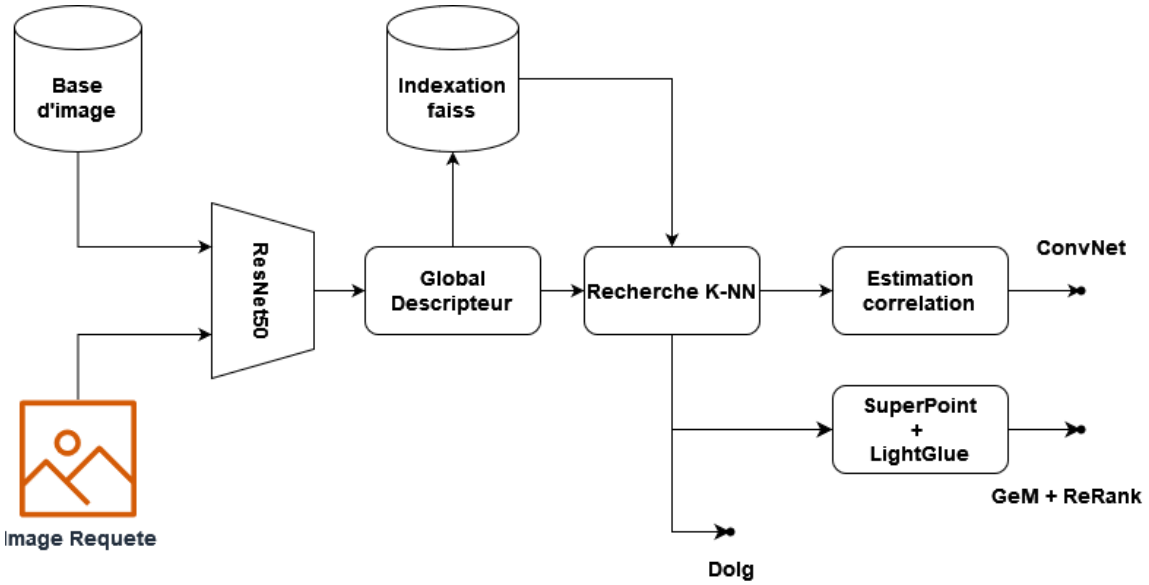


FIGURE 3.18 – Pipeline de reconnaissance d'images utilisant DOLG, GeM ou ConvNet

Afin de comparer ces différentes méthodes, nous avons employé comme encodeur le même modèle de réseau neuronal pour chacune, à savoir ResNet50. Pour la phase d'apprentissage, nous avons utilisé la base de données d'images/documents synthétiques introduite dans la Section 3.2.2. Toutes les images ont été normalisées à une résolution de 512x512 pixels pour chaque modèle et nous avons

choisi d'utiliser un optimiseur Adam. Le processus d'entraînement s'est étendu sur 60 époques, en appliquant des fonctions de perte spécifiques à chaque modèle pour s'assurer de leur concordance avec les références citées. Pour la méthode DOLG, nous avons introduit une variante de la fonction de perte décrite dans l'équation suivante :

$$loss = \lambda_1 loss_{triplet} + \lambda_2 loss_{arcface} \tag{3.3}$$

En incorporant une fonction de perte triplet, notre objectif était d'évaluer son impact sur la précision du descripteur global. Les coefficients λ_1 et λ_2 ont été utilisés pour ajuster le poids de ces deux fonctions de coût respectives. Dans notre cas, nous avons fixée ces deux valeurs à $\lambda_1 = 1$ et $\lambda_2 = 1$ Cependant, pour SuperPoint et LightGlue, nous n'avons pas entraîné les modèles sur une nouvelle base de données.

Quant à la méthodologie de test, elle suit le même protocole que celui des méthodes par description, où les images sont normalisées en termes de résolution tout en préservant le rapport original. La recherche des K-candidats est également limitée aux 40 premiers résultats.

3.4.2 Les métriques d'évaluation

La mesure de précision permet d'évaluer le nombre de prédictions positives correctes. Elle est définie comme décrit dans l'équation suivante :

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.4}$$

Plus cette valeur sera élevée, plus le nombre de faux positifs sera faible et donc plus, nous nous rapprocherons d'un système d'identification. Dans notre cas d'étude, cela correspond à déterminer si le document que le système nous fourni est correct.

La métrique de rappel vise à évaluer le pourcentage de positifs bien prédit par notre modèle. En d'autres termes, c'est le nombre de résultats positifs bien prédits (vrai positif) divisé par l'ensemble des positifs (vrai positif + faux négatif) :

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3.5}$$

Plus la mesure de rappel est élevée, plus notre système maximise le nombre de vrais positifs.

Séparément, les métriques de précision et de rappel ne permettent pas de conclure sur la performance d'un modèle. Pour avoir une mesure globale des performances, on utilise la moyenne harmonique entre ces deux valeurs, appelée F1-score :

$$F1 = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{3.6}$$

Cependant, dans notre cas, nous travaillons sur un système souhaitant se rapprocher des systèmes d'identification. Ainsi, nous souhaitons attacher une importance plus grande à la précision. La métrique F1 fait partie de la famille des F-Beta scores. Le score F1 accorde la même importance à la précision et au rappel. Nous utiliserons plutôt la métrique F-Beta qui permet d'utiliser des pondérations différentes.

$$F_\beta score = (1 + \beta^2) \frac{Recall \times Precision}{Recall + (\beta^2 . Precision)} \tag{3.7}$$

avec la valeur β pouvant modifier la pondération entre la précision et le rappel :

— Pour $\beta \geq 1$, on accorde plus d'importance au rappel

— Pour $\beta \leq 1$, on accorde plus d'importance à la précision

— Pour $\beta = 1$, on retrouve le score F1

Pour les différentes évaluations, nous fixerons la valeur $\beta = 0.25$ afin de pondérer de la précision dans la métrique d'évaluation finale du système étant donné que l'objectif est de rapprocher d'un système d'identification.

Nous évaluerons également les performances des vecteurs de description global. Pour cela, nous déterminerons la précision dite $K - NN$ que l'on notera comme la metrique $P@K$. En effet, lorsqu'une image est soumise au système pour reconnaissance, la recherche $k - NN$ permet d'identifier les k images les plus similaires (ou "proches") selon une certaine mesure de distance (par exemple, la distance euclidienne ou de hamming). La précision est dite correcte seulement lorsque le bon candidat fait partie de la liste des k candidats.

Par exemple, pour un ensemble de 100 images de requêtes, si le système $k - NN$ obtient 50 fois l'unique occurrence présente dans la base de données dans la liste des k candidats, alors la précision $P@K$ sera de 50%.

3.5 Évaluation et comparaison des méthodes

Le tableau 3.5 présente une évaluation des différentes méthodes que nous avons retenues. Cette évaluation couvre plusieurs aspects, tels que le temps d'exécution, la précision $k - NN$ (avec k=40), la précision, le rappel et le score FBeta.

Les évaluations sont présentées dans un premier temps de manière globale, puis dans un second temps par catégorie. Pour chaque méthode, les seuils ont été sélectionnés de manière à obtenir les meilleurs résultats possibles.

Premièrement, de point de vue du temps de calcul, correspondant à la recherche dans notre base d'images de références, nous constatons que les temps d'exécution sont sensiblement inférieurs pour les méthodes par apprentissage. Cette observation s'explique principalement pour deux raisons. En premier lieu, les méthodes par apprentissage bénéficient d'une accélération matérielle, c'est-à-dire qu'elles exploitent les capacités de calcul parallélisées offertes par les GPU, dans notre cas une RTX 4070. Ensuite, concernant les approches qui utilisent GeM et DOLG, une seule étape est requise. En revanche, pour ce qui est de la méthode ConvNet ou GeM avec Superpoint et LightGlue, les temps d'exécution sont d'environ 1.5 secondes. Cela s'explique par un nombre d'opérations bien plus important pour ces méthodes qui en outre est dépendant de la résolution des images. Il pourrait être possible de diminuer ces temps en imposant des résolutions plus faibles, mais au détriment de la précision.

Méthodes basées sur les descripteurs traditionnels : En se basant sur les descripteurs classiques, tels que SIFT et SURF, et en les combinant avec des méthodes d'agrégation telles que VLAD, FV et BOVW, des résultats variés ont été observés. Le VLAD combiné au SIFT a démontré une performance supérieure avec une précision de 0.71, un rappel de 0.75 et un FBeta de 0.71. En comparaison, le VLAD combiné avec SURF a obtenu une précision impressionnante de 0.88, mais un rappel plus faible de 0.57. Les méthodes FV, lorsqu'elles sont jumelées avec SIFT ou SURF, ont obtenu des scores similaires en termes de rappel, mais des variations notables en termes de précision. Les performances de

Méthode	Temps (sec)	$P@40$	Précision	Rappel	FBeta
VLAD + SIFT	0.15	0.81	0.71	0.75	0.71
VLAD + SURF	0.28	0.72	0.88	0.57	0.85
FV + SIFT	0.2	0.38	0.59	0.53	0.58
FV + SURF	0.37	0.39	0.8	0.38	0.75
BOVW + SIFT	0.14	0.16	0.19	0.53	0.2
BOVW + SURF	0.24	0.07	0.14	0.3	0.15
ResNet50 + GeM	0.09	0.86	0.54	0.44	0.53
ResNet50 + GeM + SuperPoint + Glue	1.8	0.86	0.86	0.72	0.85
ResNet50 + DOLG	0.18	0.53	0.06	0.29	0.06
ResNet50 + DOLG 2	0.2	0.59	0.18	0.45	0.18
ResNet50 + ConvNet	1.55	0.62	0.33	0.52	0.34

TABLE 3.5 – Évaluation sur l'ensemble de la base de données

BOVW, indépendamment du descripteur utilisé, étaient moins impressionnantes, avec des précisions et des scores FBeta nettement inférieurs aux autres méthodes de cette catégorie. Nous notons, de manière générale, que seule l'utilisation de VLAD permet d'avoir une précision $K - NN$ satisfaisante, ce qui joue comme nous l'observons sur les métriques suivantes.

Méthodes basées sur les réseaux neuronaux profonds : Les méthodes par apprentissage montrent une diversité dans les performances. ResNet50 couplé avec GeM offre un rappel modeste de 0.44 et une précision de 0.54. En ajoutant des techniques comme SuperPoint et LightGlue comme système de reclassement, une amélioration significative est observée avec une précision s'élevant à 0.86 et un rappel de 0.72, se traduisant par un FBeta de 0.85. La première version de DOLG a obtenu une faible précision de 0.06, bien que le rappel soit de 0.29. La seconde version, DOLG 2, avec la modification de la fonction d'apprentissage, obtient une amélioration de la précision avec un score de 0.18. L'approche ConvNnet, obtient quant à elle une précision relativement faible de 0.33, et un rappel de 0.52.

Pour synthétiser, bien que certaines méthodes, telles que VLAD associé à SIFT ou VLAD conjugué à SURF, apparaissent comme offrant un compromis idéal entre la durée d'exécution et la performance, d'autres techniques, tels que BOVW ou FV, semblent moins adaptées à ce cas d'application précis.

Dans une perspective comparative, les méthodes qui s'appuient sur des architectures profondes montrent une efficacité supérieure en ce qui concerne la recherche de $k-NN$. Ceci est particulièrement vrai lorsque GeM est configuré avec un vecteur de dimension $L = 2048$. Par ailleurs, la stratégie VLAD, largement reconnue pour sa prééminence dans les démarches descriptives, opère avec des vecteurs d'une dimension substantiellement accrue, à savoir $L = 8192$. Malgré cette grande dimensionnalité, elle délivre des performances légèrement inférieures à celles de GeM. Nous observons que la combinaison de GeM avec un système de reclassement utilisant Superpoint et LightGlue surpassent le score Fbêta atteint par VLAD en association avec SURF.

Concernant DOLG, l'introduction d'une fonction d'apprentissage modifiée semble contribuer à une amélioration des résultats par rapport à la méthode traditionnelle. De plus, malgré la dimension réduite du vecteur $L = 512$, les performances en précision $K - NN$ s'avèrent satisfaisantes. Toutefois, en matière de précision et de rappel, l'incorporation de techniques de reclassement, telles que Superpoint et LightGlue ou encore ConvNet, s'impose comme cruciale pour optimiser les résultats obtenus.

3.5.1 Catégories de documents n'ayant peu ou pas d'informations en commun

Comme expliqué précédemment, notre base de données se subdivise en diverses sous-catégories d'images. Deux grandes familles se distinguent, comprenant des images partageant des éléments substantiels, à savoir des éléments textuels et graphiques, ainsi que des ensembles d'images avec des caractéristiques visuelles plus variées. Cette section se focalise exclusivement sur les catégories (Catalogues, Affiches, Articles) qui présentent une diversité d'images.

‘ **Évaluation pour les Catalogues (Figure 3.6) :** Parmi les méthodes basées sur les descripteurs, l'approche VLAD + SURF affiche le meilleur score F-Beta de 0,98, accompagné d'une précision parfaite de 1,0, bien que le rappel soit de 0,76. Ce résultat est nettement supérieur à celui de la méthode BOVW + SURF, qui obtient le score F-Beta le plus bas de 0,22. Concernant les méthodes d'apprentissage, l'approche utilisant l'agrégateur GeM combiné à SuperPoint et LightGlue obtient les meilleurs résultats, que ce soit en termes de précision, de rappel ou, par conséquent, de score F-Beta. L'approche DOLG,

Methode	sec	$P@40$	Precision	Rappel	FBeta
VLAD + SIFT	0.15	0.91	0.81	0.77	0.81
VLAD + SURF	0.34	0.97	1.0	0.76	0.98
FV + SIFT	0.2	0.54	0.76	0.61	0.75
FV + SURF	0.41	0.7	1.0	0.6	0.96
BOVW + SIFT	0.14	0.21	0.26	0.46	0.27
BOVW + SURF	0.26	0.05	0.22	0.17	0.22
ResNet50 + GeM	0.08	0.96	0.74	0.5	0.72
ResNet50 + GeM + SuperPoint + Glue	1.75	0.96	1.0	0.83	0.99
ResNet50 + DOLG	0.18	0.24	0.02	0.19	0.02
ResNet50 + DOLG 2	0.2	0.4	0.14	0.24	0.15
ResNet50 + ConvNet	1.62	0.51	0.36	0.59	0.37

TABLE 3.6 – Évaluation pour les catalogues

Methode	sec	$P@40$	Precision	Rappel	FBeta
VLAD + SIFT	0.15	0.92	0.87	0.83	0.87
VLAD + SURF	0.27	0.91	1.0	0.82	0.98
FV + SIFT	0.2	0.84	0.91	0.78	0.9
FV + SURF	0.36	0.64	0.99	0.58	0.95
BOVW + SIFT	0.14	0.14	0.16	0.56	0.17
BOVW + SURF	0.28	0.1	0.36	0.19	0.34
ResNet50 + GeM	0.09	0.92	0.82	0.55	0.8
ResNet50 + GeM + SuperPoint + Glue	1.74	0.92	0.99	0.91	0.99
ResNet50 + DOLG	0.18	0.65	0.14	0.2	0.14
ResNet50 + DOLG 2	0.19	0.62	0.38	0.43	0.38
ResNet50 + ConvNet	1.65	0.84	0.7	0.81	0.71

TABLE 3.7 – Évaluation pour les Affiches

avec une fonction d'apprentissage modifiée, surpasse la version utilisant une fonction d'apprentissage unique, mais obtient tout de même des résultats relativement faibles. En ce qui concerne ConvNet, les résultats sont peu satisfaisants, tout en nécessitant un temps d'exécution considérablement plus long.

Évaluation pour les Affiches (Figure 3.7) : Les scores sont légèrement différents dans cette catégorie. Encore une fois, "VLAD + SURF" excelle avec un score FBeta de 0.98, grâce à sa précision parfaite de 1.0 et un rappel de 0.82. C'est nettement mieux que "BOVW + SIFT" qui a le score FBeta le plus bas de 0.17. Pour les méthodes basées sur l'apprentissage, l'ordre est sensiblement le même, avec GeM obtenant les meilleurs résultats, que ce soit avec reclassement ou non par rapport aux autres méthodes par apprentissages. Nous notons, cependant, que la méthode ConvNet a montré une amélioration significative comparée aux catalogues avec un score FBeta compétitif de 0.71. Cela peut s'expliquer par la nature même des informations contenues sur les affiches. En effet, ces images sont principalement composées d'informations graphiques et non textuels.

Évaluation pour les Articles (Figure 3.8) : Les performances dans cette catégorie sont quelque peu variées. La méthode VLAD + SIFT affiche une performance robuste avec un score F-Beta de 0,84.

Methode	sec	$P@40$	Precision	Rappel	FBeta
VLAD + SIFT	0.16	0.77	0.85	0.78	0.84
VLAD + SURF	0.43	0.53	0.75	0.37	0.71
FV + SIFT	0.2	0.63	0.89	0.66	0.87
FV + SURF	0.53	0.33	0.77	0.27	0.69
BOVW + SIFT	0.14	0.4	0.52	0.57	0.52
BOVW + SURF	0.26	0.04	0.05	0.26	0.06
ResNet50 + GeM	0.09	0.92	0.67	0.41	0.64
ResNet50 + GeM + SuperPoint + Glue	1.65	0.92	0.99	0.85	0.98
ResNet50 + DOLG	0.18	0.49	0.1	0.02	0.08
ResNet50 + DOLG 2	0.19	0.58	0.58	0.16	0.5
ResNet50 + ConvNet	1.69	0.51	0.45	0.26	0.43

TABLE 3.8 – Évaluation pour les articles

Ce résultat est nettement supérieur à celui de BOVW + SURF, qui obtient encore une fois le score F-Beta le plus bas de 0,06. En ce qui concerne les méthodes basées sur l'apprentissage, la précision en $k - NN$ est encore une fois très bonne, surtout pour le cas de GeM. On note, le score Fbeta le plus haut revient encore une fois à la combinaison GeM + SuperPoint + LightGlue.

En conclusion, la combinaison GeM + SuperPoint + Lightglue offre les meilleurs résultats, mais un temps de processus relativement long. les approches par descriptions, telles que VLAD avec SIFT ou SURF, offrent des résultats relativement haut également, mais avec un temps d'exécution nettement plus faible. Il semblerait que les approches utilisant un système de reclassement basé sur des points d'intérêt et obtenant des scores $K - NN$ relativement haut, soit la meilleure solution et ne semble pas souffrir de gros problèmes de confusion pour ces catégories d'images.

3.5.2 Problématiques de confusion pour les images partageant des informations

Les résultats qui suivent mettent en évidence l'efficacité de diverses méthodologies de reconnaissance d'images appliquées à trois catégories distinctes : cartes de visite, factures et modèles. Il est important de noter que ces catégories d'images se caractérisent par la présence d'informations textuelles et graphiques similaires ou de mise en page identiques. Ces éléments variés ont le potentiel d'introduire une source de confusion dans le processus de reconnaissance qui est l'une des principales problématiques dans le cadre industriel.

Évaluation cartes de visites (Figure 3.9) : Pour cette première catégorie, contenant des images ayant un nombre restreint d'informations et une mise en page identique, la méthode VLAD + SIFT s'est révélée être la plus performante en ce qui concerne la mesure $k - NN$, avec une valeur de 0,73. Cependant, la combinaison VLAD + SURF, obtient des résultats supérieurs pour la précision malgré un rappel inférieur avec un score FBeta de 0.67, soit le plus haut score.

Ensuite, les approches basées sur l'apprentissage ont obtenu des résultats homogènes pour la précision $K - NN$ variant de 0.34 à 0.55. Cependant, la précision finale maximale obtenue est de seulement 0.48 dans le cas de GeM avec reclassement. Nous notons une première difficulté pour la recherche $K-NN$ ainsi que pour la précision avec une confusion importante et une difficulté à identifier correctement l'image au sein de la base de données.

Évaluation factures (Figure 3.10) : Pour la catégorie des factures, nous observons une augmentation notable des résultats sur la recherche, $K - NN$ que ce soit pour les méthodes par description ou par apprentissage. Cependant, même pour les approches les plus performantes, utilisant un système de reclassement, la précision finale témoigne d'une confusion. Malgré une augmentation des seuils de validation, la précision ne parvient pas obtenir un score parfait.

Évaluation modèle d'affiches (Figure 3.11) : Concernant cette catégorie, il est notable que les scores obtenus via la méthode des k plus proches voisins $(K - NN)$ se révèlent relativement satisfaisants pour une majorité des méthodes employées. Cette première conclusion semble principalement attribuable à la faible quantité d'images au sein de chaque sous-catégorie. Il est conjecturable qu'en présence de groupes de documents de l'envergure des factures ou des cartes de visite, ces résultats pourraient subir une baisse notable.

Il est par, ailleurs, observable que la précision et le rappel posent à nouveau un problème, témoignant ainsi d'une confusion. En dépit de l'emploi de méthodes de reclassement, les divers pipelines ne par-

Méthode	sec	$P@40$	Précision	Rappel	FBeta
VLAD + SIFT	0.11	0.73	0.62	0.72	0.62
VLAD + SURF	0.12	0.32	0.7	0.4	0.67
FV + SIFT	0.17	0.04	0.09	0.4	0.09
FV + SURF	0.21	0.01	0.11	0.05	0.1
BOVW + SIFT	0.12	0.03	0.04	0.5	0.04
BOVW + SURF	0.15	0.11	0.12	0.52	0.13
ResNet50 + GeM	0.09	0.54	0.13	0.09	0.13
ResNet50 + GeM + SuperPoint + Glue	1.98	0.54	0.48	0.5	0.48
ResNet50 + DOLG	0.19	0.34	0.01	0.67	0.01
ResNet50 + DOLG 2	0.19	0.55	0.02	0.6	0.02
ResNet50 + ConvNet	0.72	0.46	0.02	0.46	0.02

TABLE 3.9 – Évaluation pour les cartes de visites

Méthode	sec	$P@40$	Précision	Rappel	FBeta
VLAD + SIFT	0.16	0.91	0.83	0.83	0.83
VLAD + SURF	0.17	0.72	0.92	0.65	0.9
FV + SIFT	0.2	0.03	0.09	0.3	0.09
FV + SURF	0.26	0.11	0.42	0.25	0.4
BOVW + SIFT	0.14	0.01	0.01	0.49	0.01
BOVW + SURF	0.19	0.05	0.1	0.37	0.1
ResNet50 + GeM	0.09	0.79	0.22	0.34	0.23
ResNet50 + GeM + SuperPoint + Glue	1.93	0.79	0.83	0.87	0.83
ResNet50 + DOLG	0.18	0.79	0.12	0.56	0.13
ResNet50 + DOLG 2	0.18	0.62	0.06	0.76	0.06
ResNet50 + ConvNet	1.77	0.23	0.03	0.13	0.03

TABLE 3.10 – Évaluation pour les factures

Méthode	sec	$P@40$	Précision	Rappel	FBeta
VLAD + SIFT	0.16	0.72	0.36	0.64	0.37
VLAD + SURF	0.28	0.79	0.77	0.52	0.75
FV + SIFT	0.2	0.15	0.13	0.41	0.14
FV + SURF	0.36	0.43	0.6	0.44	0.58
BOVW + SIFT	0.14	0.1	0.06	0.56	0.07
BOVW + SURF	0.27	0.07	0.13	0.34	0.13
ResNet50 + GeM	0.09	0.97	0.38	0.62	0.39
ResNet50 + GeM + SuperPoint + Glue	1.8	0.97	0.65	0.45	0.63
ResNet50 + DOLG	0.19	0.71	0.06	0.29	0.06
ResNet50 + DOLG 2	0.22	0.72	0.18	0.62	0.19
ResNet50 + ConvNet	1.7	0.89	0.17	0.7	0.18

TABLE 3.11 – Évaluation pour les modèles d'affiches

viennent pas à identifier de façon précise l'occurrence adéquate présente dans la base de données.

Pour ces classifications d'images, une décroissance globale de la précision en $k - NN$ est observée. Une diminution substantielle de la précision en $1 - NN$ est également notable, signalant une exacerbation du phénomène de confusion. Il convient de rappeler que cela suggère que le système de reconnaissance d'images est susceptible de fournir des résultats fallacieux, tout en les considérant comme exacts.

Dans le contexte spécifique des factures et des cartes de visite, où les informations sont principalement textuelles, les méthodes fondées sur l'apprentissage automatique semblent nettement surpassées par la méthode VLAD. Inversement, lorsque les informations sont principalement graphiques et de dimensions supérieures, les méthodes axées sur l'apprentissage démontrent une supériorité. Il est aussi pertinent de souligner que, malgré l'incorporation de méthodes de reclassement, la précision semble diminuer comparativement aux catégories d'images initiales.

3.6 Bilan

Dans ce chapitre, nous avons tout d'abord analysé les bases de données existantes, mettant en évidence les problématiques en matière de reconnaissance et d'identification d'images à partir de dispositifs tels que des caméras. Les limitations identifiées ont servi de moteur pour la création d'une base de données spécifiquement adaptée à notre problématique. L'intégration d'une méthode de génération de données synthétiques s'est avérée indispensable pour affiner notre ensemble de données, en particulier pour entraîner les méthodes basées sur l'apprentissage profond. Afin d'assurer une évaluation rigoureuse, un ensemble de métriques a été proposé pour l'évaluation de différentes méthodes.

Ce chapitre a également servi de cadre pour l'exploration et l'évaluation de diverses techniques de reconnaissance d'images. Qu'il s'agisse de méthodes descriptives telles que SIFT, SURF, VLAD, Fisher Vector et BOVW, ou de méthodes basées sur l'apprentissage comme GeM, SuperPoint, LightGlue, DOLG et ConvNet, nos analyses initiales ont permis de déterminer la combinaison optimale de méthodes descriptives, à savoir VLAD associé à SIFT/SURF ou GeM associe a SuperPoint et LightGlue. De plus, il a été observé que la fonction d'apprentissage joue un rôle essentiel, permettant d'améliorer significativement les résultats sans altérer l'architecture sous-jacente. À titre d'exemple, alors que GeM génère des vecteurs de dimension $L = 2048$, DOLG 2 produit des résultats intéressants avec des vecteurs de dimension $L = 512$.

En ce qui concerne l'approche ConvNet, elle semble moins adaptée à notre problématique et affiche globalement des performances inférieures pour les méthodes basées sur l'apprentissage, malgré sa complexité accrue. Cependant, il est envisageable que des améliorations puissent être obtenues avec une base de données d'apprentissage plus vaste, comprenant un plus grand nombre de données de référence.

Il est également important de noter que les images partageant des éléments graphiques ou textuels ou simplement une mise en page similaire tendent à augmenter les risques de confusion. Cette problématique est particulièrement importante dans le contexte des applications de réalité augmentée. De plus, une diminution des mesures de rappel a été observée pour les images peu informatives ou de petites tailles telles que les cartes de visites. Par conséquent, il est plausible de supposer que la présence accrue d'éléments d'arrière-plan pourrait avoir un impact négatif sur les recherches $K - NN$

et donc sur la précision et le rappel.

Chapitre 4

Système de reconnaissance d'image proposé

Ce chapitre introduit un tout nouveau modèle de détection de documents, spécifiquement conçu et optimisé pour une utilisation dans les navigateurs web et sur les appareils à faible puissance. Reconnaissant les contraintes de performance et de ressources de tels environnements, ce modèle vise à offrir une détection rapide et précise sans sacrifier la qualité. Par la suite, le chapitre dévoile un système innovant de recherche pour la reconnaissance d'images, offrant une amélioration significative en termes de précision. Cette avancée s'annonce comme une étape cruciale pour améliorer l'efficacité des applications de reconnaissance d'images, indispensable pour les applications de réalité augmentée.

4.1 Introduction

Comme démontré au chapitre précédent, les méthodologies actuelles rencontrent des difficultés à parvenir à un niveau de précision satisfaisant. Indépendamment de leur fondement, qu'il soit à base d'apprentissage statistique ou de descripteurs visuels, des incertitudes persistent, en particulier lorsque les images comportent des éléments d'apparence similaire. De plus, la présence d'éléments en arrière-plan tend à diminuer l'efficience du système en matière de rappel, engendrant potentiellement la frustration de l'utilisateur en raison de multiples essais de reconnaissance non concluants.

Dans une première partie, notre attention se porte sur les méthodes relatives à la détection de documents. Ces techniques de segmentation visent à séparer de manière distincte les documents de leur contexte d'apparition. Une telle démarche est fréquemment adoptée dans les systèmes de reconnaissance optique de caractères (OCR). Notre objectif est de concevoir une solution à la fois performante et facilement intégrable dans les outils des utilisateurs, telles que les applications Web, et d'évaluer son impact sur des systèmes de reconnaissance.

Dans la seconde partie, nous introduisons une nouvelle méthodologie pour la reconnaissance d'images planes, axée principalement sur la détection de sous-parties d'images. L'objectif primordial est de minimiser la redondance des informations stockées dans la base de données tout en optimisant la précision et le rappel grâce à une technique de recherche par fenêtrage.

4.2 Segmentation et détection de documents

Les méthodologies conventionnelles de création de descripteurs, qu'ils soient locaux ou globaux, sont souvent compromises par la présence d'informations superflues, notamment les éléments en arrière-plan. Cette vulnérabilité devient manifeste lorsque les documents sont dépourvus d'attributs distinctifs, telles que des fréquences élevées.

Dans ce cadre, notre objectif est d'élaborer un système capable de détecter et éventuellement de rectifier la région correspondant au document, afin de produire une image optimisée. Des études antérieures ont démontré les avantages inhérents d'une telle approche, comme en témoignent les travaux de [Zha+17; Xue+22]. En outre, notre objectif est de créer un système opérant en mode client, idéalement sous la forme d'une application Web, afin de réduire la charge de calcul imposée au serveur. Une telle initiative induit des enjeux particuliers en matière d'architecture et de gestion des ressources, notamment dans le contexte des applications Web.

Une tendance se distingue dans l'état de l'art. Les recherches actuelles favorisent majoritairement les approches par apprentissage, qui se sont révélées être d'une efficacité supérieure, particulièrement lorsqu'il s'agit de traiter des images caractérisées par des arrière-plans complexes ou des entraves visuelles.

Ainsi, nous avons choisi d'adapter l'architecture HuPageScan pour élaborer notre propre mécanisme de détection de documents. Bien que certaines études telles que LDRNet [Wu+23] offrent certains bénéfices, le nombre élevé de leurs paramètres et la complexité relative à leurs couches entièrement connectées les rendent moins appropriées pour les applications Web. Pour pallier ces inconvénients, nous avons privilégié des approches par auto-encodeurs dans le but de minimiser le nombre de paramètres et d'opérations associées. Cette méthodologie nous permet d'espérer une efficacité accrue dans des dispositifs ou des contextes caractérisés par des ressources de calcul limitées.

4.2.1 Évaluation et protocole d'expérimentation

Nous avons retenu la base de données SmartDoc pour évaluer la performance de notre système [Bur+15] présenté dans la section 3.2.1.3. Cette base a été divisée en deux sous-ensembles : 20 % pour l'entraînement du modèle et les 80 % restants pour les tests. Afin d'enrichir notre ensemble d'entraînement, nous avons généré 15 000 images supplémentaires grâce au mécanisme de génération de données synthétiques décrit dans la section 3.3.

Comme métrique des performances, nous avons retenu l'indice de Jaccard, couramment appelé Intersection sur Union (IoU). Cette métrique évalue la concordance entre la vérité terrain et nos prédictions en utilisant le rapport entre l'intersection et l'union de deux ensembles.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.1}$$

Pour chaque expérimentation, les diverses architectures ont été entraînées en utilisant la base de données susmentionnée, avec des images d'une résolution de (224 x 224) pixels. Il est à noter que cette résolution relativement faible, bien qu'entraînant une légère perte de précision par rapport aux images en résolution native, présente l'avantage de diminuer le temps de calcul nécessaires

Durant la phase d'entraînement, nous avons employé la fonction de coût DiceLoss et un optimiseur Adam avec un taux d'apprentissage initial de 0,0001. Ce taux est ajusté tous les 15 cycles, ou epochs. Nos architectures traitent des lots comprenant 6 images, avec un total de 80 époques pour la durée de l'entraînement. Cette approche intègre à la fois les erreurs locales et globales, ce qui est crucial pour garantir une précision optimale.

4.2.2 Méthodologie proposée : l'architecture FastNet

L'architecture FastNet [LPZ22b] que nous avons proposée est spécifiquement optimisée pour la segmentation de documents acquis par caméra pour des applications Web.

4.2.2.1 Influence de la profondeur du réseau sur la performance

Afin de développer une architecture à la fois efficace et peu coûteuse en termes de complexité calculatoire, nous avons initié notre étude en analysant l'impact de la profondeur d'un réseau sur la précision de la segmentation. Pour cela nous avons retenu l'architecture HU-PageScan [Nev+20], une déclinaison de U-Net [RFB15], qui se distingue par une diminution progressive du nombre de filtres de convolution à chaque niveau, comme illustré dans la figure 4.2.

Nous avons donc entamé nos expérimentations avec le modèle U-Net de base, illustré dans la figure 4.1. La motivation derrière ce choix réside dans les performances exceptionnelles observées sur la base de données SmartDoc [Bur+15].

À partir du schéma de Hu-PageScan, deux architectures distinctes ont été formulées, se distinguant essentiellement par le nombre de convolutions employées. La première dénomination, UNetLight 1, présente une réduction de quatre convolutions, tandis que la seconde, UNetLight 2, est caractérisée par une réduction de huit convolutions. Ces modifications impactent directement le nombre global de paramètres (avec, respectivement, 1 928 417 et 467 808 paramètres pour UNetLight 1 et 2) et le volume d'opérations.

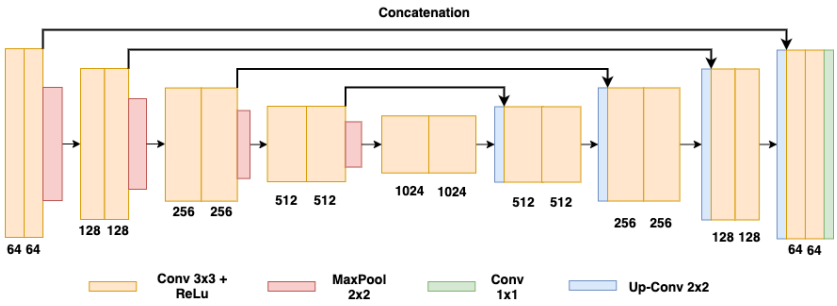


FIGURE 4.1 – U-Net Architecture : 31 043 521 paramètres [RFB15]

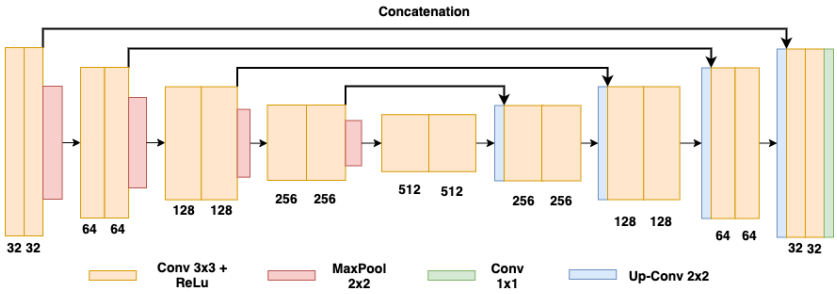


FIGURE 4.2 – HU-PagseScan Architecture : 7 765 985 paramètres [Nev+20]

Nos résultats préliminaires, rapportés dans le Tableau 4.1, soulignent les conséquences de la réduction de convolutions sur la précision de segmentation. Dans les quatre premiers contextes, marqués par des arrière-plans simples, l'écart de performance entre les architectures est négligeable. Toutefois, face à un arrière-plan hétérogène avec des occlusions, comme illustré dans le cinquième contexte, la performance de l'architecture UnetLight 2 est très légèrement diminuée.

À la lumière de ces observations, nous avons opté pour une exploration approfondie de l'architecture UnetLight 2. Comme en attestent les résultats du Tableau 4.1, cette architecture, bien que présentant une précision comparable à celle des modèles U-Net et HU-PageScan, offre l'avantage d'une réduction significative du nombre de paramètres, qui est diminué de 36 fois par rapport à U-Net et d'un temps d'inférence inférieur.

4.2.2.2 Modification du codeur et du décodeur

L'architecture UNetLight 2 constitue le socle sur lequel nous avons bâti nos améliorations. Ces dernières ont été articulées autour de deux dimensions majeures : l'encodage et le décodage. Notre objectif premier visait à remodeler cette architecture en vue d'optimiser la vitesse d'exécution, en réduisant le nombre total d'opérations requises.

En ce qui concerne l'encodage, nous avons opté pour une modification de la seconde couche convolutionnelle en y introduisant un pas de déplacement (ou "stride") de 2, ce qui a pour conséquence directe de diminuer la résolution. Cette opération offre l'avantage de s'affranchir de la nécessité d'une opération de "max pooling". Notons que cette approche est en concordance avec les méthodologies mises en œuvre lors des premières étapes convolutionnelles des architectures ResNet [He+16]. Cette modification conduit à une division par deux du nombre d'opérations, ce qui engendre une accélération

Méthodes	Paramètres	Temps	Fond 1	Fond 2	Fond 3	Fond 4	Fond 5	Avg
Unet	17267393	0.82	**0.9905**	**0.9891**	**0.9913**	**0.9891**	**0.9869**	**0.9898**
Notre HuPageScan	7765985	0.31	0.9900	0.9883	0.9908	0.9883	0.9853	0.9891
UnetLight 1	1928417	0.24	0.9900	0.9884	0.9909	0.9883	0.9850	0.9891
UnetLight 2	**467809**	**0.18**	0.9895	0.9878	0.9899	0.9868	0.9777	0.9876

TABLE 4.1 – Comparaison des différents modèles sur la base de données SmartDOC

notable de la phase d'encodage. Il est intéressant de souligner que cette modification n'a qu'un impact minime sur la précision, avec une baisse approximative de 0,002 en termes de IoU.

En ce qui concerne la phase de décodage, l'enjeu restait le même : comment optimiser la vitesse d'inférence tout en garantissant une précision adéquate. Notre premier obstacle technique a été de compenser la diminution de la taille des champs récepteurs, résultant de l'omission de certaines couches convolutionnelles dans la phase d'encodage. À cette fin, nous avons puisé notre inspiration du modèle FastFCN [Wu+19], qui préconise l'utilisation de couches convolutionnelles en parallèle, chacune ayant des niveaux de dilatation distincts. Cette technique offre une solution robuste pour traiter les problématiques associées à la taille de champs récepteurs. Les cartes de caractéristiques issues de différentes échelles ont été amalgamées et traitées via ce système convolutionnel.

Dans la continuité de ces améliorations, nous avons intégré à notre architecture un module combinant l'attention spatiale et l'attention par canal. Ce module ambitionne de reproduire la manière dont la vision humaine traite l'information : de façon sélective et séquentielle, plutôt que globale. Divers travaux antérieurs ont préconisé l'intégration de tels mécanismes d'attention pour amplifier l'efficacité des réseaux neuronaux convolutionnels dans les tâches de classification [Wan+17; HSS18]. Cette attention se divise principalement en deux catégories : l'attention spatiale et l'attention par canal, correspondant respectivement aux notions de "où" et de "quoi".

4.2.2.2.1 Attention spatiale Le module d'attention spatiale autorise le réseau à privilégier les régions pertinentes de l'image selon leur contexte spatial. Cette démarche implique l'attribution d'un poids à chaque pixel de l'image, calibré en fonction de sa pertinence relative à la tâche spécifiée. De façon plus détaillée, une carte de poids est élaborée sur la base des caractéristiques dérivées de l'image à travers chaque strate du réseau neuronal. Subséquemment, cette carte des poids est mise en correspondance avec l'image initiale afin de produire une version pondérée de celle-ci. Dans cette transformation, les pixels jugés pertinents sont mis en exergue tandis que ceux de moindre importance sont modulés à la baisse. L'image résultante, ajustée par le processus d'attention, est ensuite relayée aux strates ultérieures du réseau pour les phases d'analyse ultérieures.

4.2.2.2.2 Attention par cannal Les systèmes d'attention par canal, aussi appelés attention channel-wise, autorisent le réseau à privilégier certaines caractéristiques spécifiques au sein des cartes de caractéristiques produites par les couches convolutionnelles. Cette modalité opère en générant une carte de pondération qui est par la suite appliquée à la carte de caractéristiques, permettant ainsi d'exacerber ou de modérer certains canaux spécifiques.

Le module que nous avons incorporé à la sortie de notre système convolutionnel dilaté intègre ces deux mécanismes d'attention, permettant ainsi au réseau de déterminer non seulement "quoi" (examiner), mais également "où" (focaliser) son attention. L'agencement séquentiel de ces deux modules d'attention s'avère optimal, comme en témoignent des études ayant proposé un module analogue [Woo+18] CBAM (*Convolutional Block Attention Module*.

Postérieurement aux modules convolutionnels dilatés, un module d'attention a été intégré (CBAM), recevant en entrée la sortie de ce dernier ainsi que les informations de l'encodeur. Les résultats issus de ce module sont ensuite fusionnés avec les sorties du module convolutionnel et transmis à travers des couches convolutionnelles visant à améliorer la résolution de l'image. En fin de compte, ces informations sont combinées avec celles émanant de la première couche convolutionnelle, puis elles transitent

Méthodes	Paramètres	Temps	Fond 1	Fond 2	Fond 3	Fond 4	Fond 5	Avg
HuPageScan [Nev+20]	7765985	0.31	-	-	-	-	-	0.9923
Notre HuPageScan	7765985	0.31	0.9900	0.9883	0.9908	0.9883	0.9853	0.9891
FastNet	**768099**	**0.08**	0.9897	0.9879	0.9904	0.9874	0.9806	0.9882
LDRNet [Wu+23]	-	-	0.9877	0.9838	0.9862	0.9802	0.9858	0.9849
SEECS-NUST-2 [JS17]	-	-	0.9832	0.9724	0.9830	0.9695	0.9478	0.9743
LRDE [Bur+15]	-	-	0.9869	0.9775	0.9889	0.9837	0.8613	0.9716

TABLE 4.2 – Comparaison des différents modèles et de l'état de l'art sur la base de données SmartDOC

à travers une dernière couche de convolution, aboutissant à la sortie envisagée. Une représentation schématique de l'architecture proposée, nommée FastNet, est illustrée Figure 4.3.

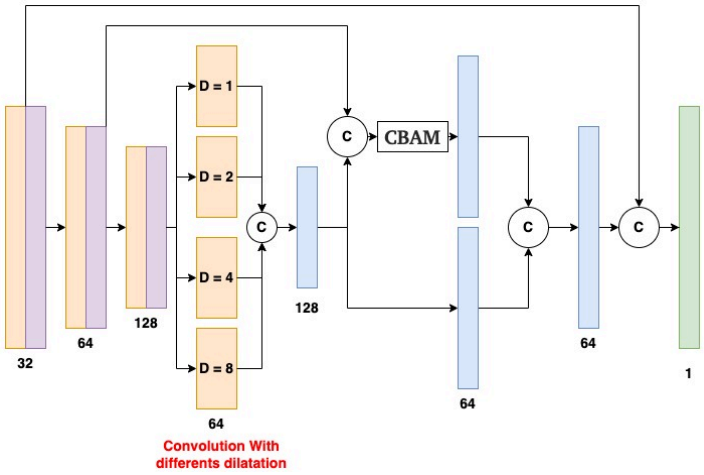


FIGURE 4.3 – Architecture FastNet : 768 099 paramètres

Les résultats préliminaires, présentés dans le Tableau 4.2, attestent clairement que les améliorations apportées influencent positivement à la fois le poids du modèle et la vitesse d'inférence. En adoptant notre architecture, la vitesse d'inference de bout en bout s'est accrue de 2,25 fois, tout en affichant une précision supérieure comparée à celle d'UNetLight 2.

Notre architecture FastNet présente une performance accrue par rapport à Unet-Light 2, bien qu'elle comporte davantage de paramètres. Lorsque comparé à Hu-PageScan [Nev+20], FastNet arbore un nombre de paramètres réduit par un facteur de 10,1, une vitesse d'inférence de bout en bout améliorée par un facteur de 3,8, le tout avec une différence de précision négligeable (moins de 0,01).

Dans l'état actuel, selon le Tableau 4.2, notre méthode occupe la deuxième position en matière de précision. Comparativement à d'autres approches, il reste complexe de juger de la rapidité de notre modèle, sauf en ce qui concerne Hu-PageScan et U-Net, pour lesquels une approche semblable a été mise en œuvre. Néanmoins, il est manifeste que FastNet dépasse ces méthodes en termes de rapidité. En conclusion, FastNet délivre des performances en adéquation avec Hu-PageScan, mais avec un avantage distinct en termes de rapidité et de compacité du modèle.

4.2.2.3 Expérimentations avec l'application Web

Ces travaux autour de l'élaboration d'un modèle de segmentation et de détection de documents ont visé à concevoir un système opérationnel sur des dispositifs à faible puissance, en particulier dans un contexte web.

À cet égard, la bibliothèque TensorFlow.js s'avère particulièrement pertinente pour utiliser des modèles au sein de navigateur. Il s'agit d'une bibliothèque JavaScript conçue par l'équipe TensorFlow de Google, permettant la formation et l'exécution de modèles d'apprentissage automatique directement au sein du navigateur web. Exploitant WebGL, TensorFlow.js parvient à mobiliser l'accélération graphique d'un navigateur pour effectuer des opérations tensorielles. Cela offre la possibilité de déployer des modèles par apprentissage profond en direct dans le navigateur, éliminant le besoin d'un serveur back-end.

De plus, il est envisageable de transposer des modèles pré-entraînés TensorFlow (écrits en Python) en format TensorFlow.js, facilitant leur mise en œuvre dans le navigateur. Ce mécanisme rend le déploiement des modèles relativement simple.

Cependant, nos recherches ont été intégralement menées en utilisant le framework PyTorch. Par conséquent, il nous a été nécessaire d'élaborer initialement un pipeline, illustré dans la Figure 4.4, pour convertir nos modèles [Pas+19]. La première étape de ce processus a impliqué la conversion du modèle au format ONNX (Open Neural Network Exchange Format, un standard destiné à représenter n'importe quel modèle d'apprentissage automatique ou profond). Par la suite, le modèle a été transposé en format TensorFlow [Aba+16], pour finalement être adapté à une utilisation via la bibliothèque TensorFlow.js.

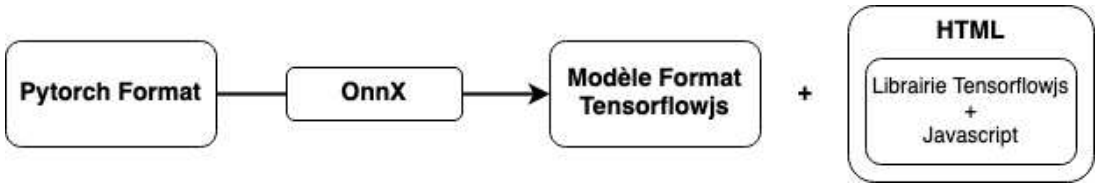
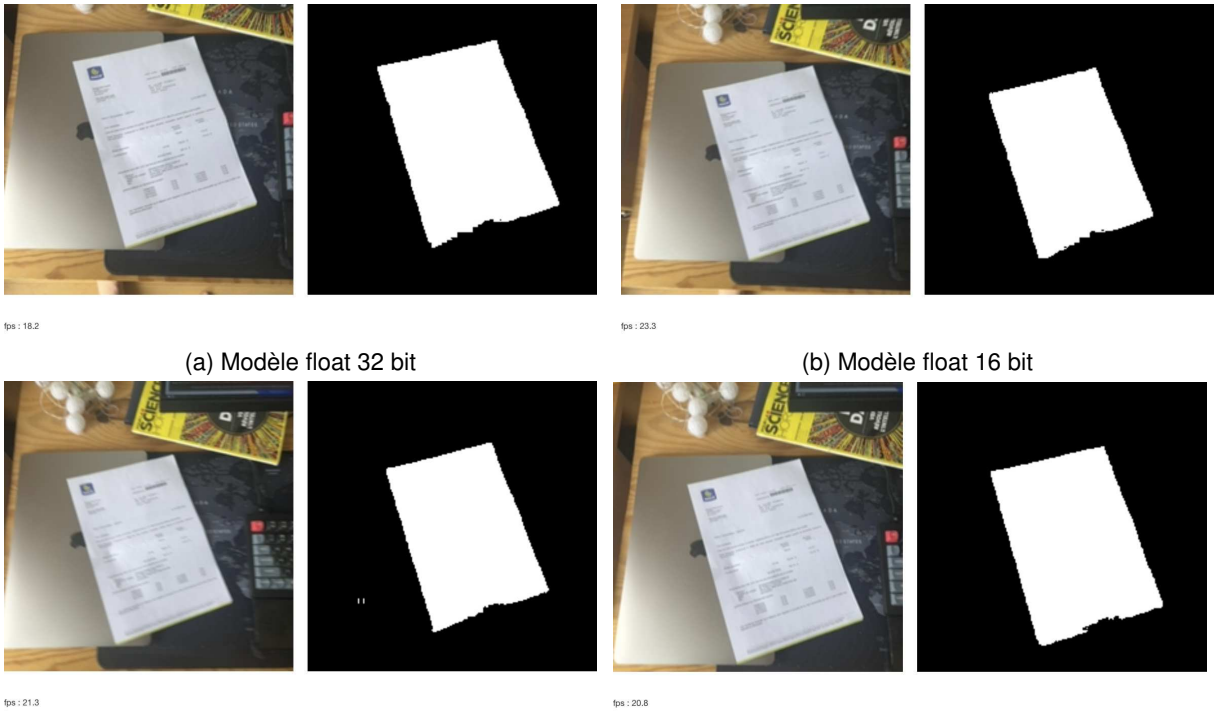


FIGURE 4.4 – Schéma de passage du format Pytorch puis à son utilisation dans une application Web

Lors de la conversion d'un modèle, pour qu'il soit compatible avec TensorFlow.js, il est également possible de recourir à des techniques de quantification afin de minimiser la taille du modèle en question. La quantification est une méthode qui consiste à réduire la précision des poids, des activations, ainsi que d'autres paramètres inhérents aux modèles par apprentissage profond. L'objectif principal de cette démarche est triple : diminuer la taille globale du modèle, augmenter la rapidité d'inférence et minimiser la consommation énergétique. Et ce, tout en s'efforçant de préserver, dans la mesure du possible, la précision intrinsèque du modèle. Cette technique s'avère particulièrement utile pour les dispositifs à ressources limitées, tels que les appareils mobiles ou les environnements web.

Les poids d'un réseau de neurones sont typiquement représentés en tant que nombres en virgule flottante de 32 bits (float32). Toutefois, dans de nombreux cas, ces poids peuvent être représentés avec une précision moindre, telle que float16, uint16 ou même uint8, sans engendrer de perte significative en matière de précision du modèle.

Dans le cadre de notre étude, nous avons expérimenté quatre variantes de modèles quantifiés, dont les performances ont été évaluées sur un iPhone 13 via le navigateur Safari (comme le détaille le tableau 4.3). Nos résultats préliminaires indiquent une capacité de traitement d'environ 20 à 23 images par seconde sur cet appareil. Cependant, lorsque nous testons ces mêmes modèles sur un dispositif



(a) Modèle float 32 bit

(b) Modèle float 16 bit

(c) Modèle uint 16 bit

(d) Modèle uint 8 bit

FIGURE 4.5 – Exemple de résultats de notre modèle avec différents niveaux de quantification, float32, float16, uint16 et uint8.

technologiquement antérieur, tel que l'iPhone 8, la performance diminue de moitié, en affichant un taux de 10 à 16 images traitées par seconde, suivant le type d'architecture. Le principal intérêt, dans notre cas, réside tout de même dans la diminution du poids de l'architecture à charger à chaque utilisation de l'application Web (Tableau 4.3).

Quantification	Float32	Foat16	Uint16	Uint8
Poids	4.3 Mo	2.2 Mo	2.2 Mo	1.1 Mo

TABLE 4.3 – Quantification de notre modèle Pytorch au format TensorflowJS

Il est important de noter que la quantification peut entraîner une perte de précision du modèle. Cependant, selon le modèle et l'application, cette perte peut être négligeable. Nous remarquons, que dons notre cas, les résultats et la précision sont relativement similaires (Figure 4.5). De plus, nous avons testé ces 4 modèles une nouvelle fois sur notre base de données, avec une diminution de la précision allant jusqu'à 0.11 dans le cas de uint8.

4.2.3 Intégration dans le pipeline du système de reconnaissance

Notre ambition principale réside dans l'exploitation de ces systèmes de pré-traitement principalement côté client (sur l'appareil de l'utilisateur), afin de minimiser la sollicitation des serveurs. Cette démarche se concrétise grâce à la transition vers le format TensorFlow.js, facilitant l'exécution au sein d'un navigateur web.

Une fois la prédiction obtenue, le masque de segmentation s'avère essentiel, focalisant l'analyse sur une zone précise de l'image. Toutefois, pour pallier les erreurs éventuelles de prédiction, nous avons intégré diverses étapes afin d'assurer une délimitation adéquate de cette zone et de réduire les éventuelles altérations de l'image en cas de prédictions défaillantes.

Effectivement, dans certaines situations, les prédictions peuvent s'avérer non pertinentes, comme illustrées dans la Figure 4.6. Il devient alors impératif de ne pas se fier à ces prédictions, car une mauvaise évaluation de la zone d'intérêt pourrait nuire considérablement à la reconnaissance ultérieure de l'image.

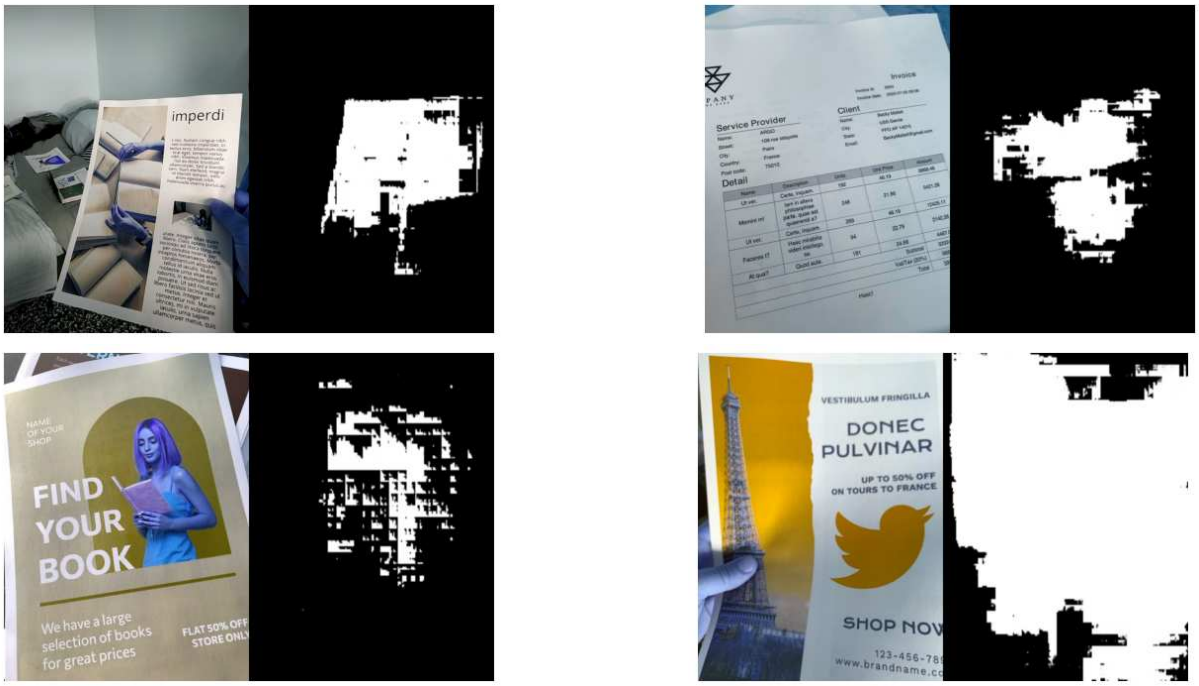


FIGURE 4.6 – Exemple de mauvaises prédictions du système de segmentation

Afin d'évaluer la justesse de la prédiction, nous avons intégré plusieurs étapes de traitement d'images. L'ambition centrale est de vérifier la cohérence du masque prédit en s'assurant qu'il évoque une forme généralement assimilable à un document, typiquement un parallélogramme.

La démarche initiale implique l'application d'opérations de morphologie mathématique sur le masque. En utilisant des noyaux spécifiques, nous réalisons une ouverture morphologique (érosion suivie d'une dilatation) pour éliminer les éventuels artefacts de prédiction. Par la suite, une détection de contours est effectuée afin d'isoler la forme dominante au sein du masque, selon la méthodologie de Suzuki [Suz+85]. C'est cette forme qui servira de base pour évaluer la cohérence du masque. Plus précisément, nous cherchons à déterminer si la forme obtenue peut être qualifiée de parallélogramme.

Dans cette optique, suite à la détection de contours, nous mettons en œuvre une approximation de la forme. L'idée est de représenter une courbe ou un polygone par une autre courbe ou polygone comportant un nombre réduit de sommets, tout en garantissant que leur écart respecte une précision prédéfinie. Cette étape s'appuie sur l'algorithme de Douglas-Peucker [NV96]. Une fois les quatre arêtes dominantes de la forme identifiées, nous évaluons les angles formés entre ces arêtes, permettant ainsi de confirmer ou d'invalider la prédiction comme étant un document.

FIGURE 4.7 – Résultat du pipeline de segmentation jusqu'à estimation de la zone d'intérêt

Catégories	Précision	Rappel	$P@40$	Fbeta
Avg	0.92(+0.04)	0.67(+0.11)	0.9(+0.19)	0.9(+0.05)
Carte de Visite	0.96(+0.21)	0.56(+0.17)	0.61(+0.26)	0.92(+0.21)
Catalogue	0.99(-0.01)	0.81(+0.06)	1.0(+0.03)	0.98(+0.0)
Facture	0.97(+0.05)	0.81(+0.21)	0.91(+0.23)	0.96(+0.07)
Affiche	0.99(-0.01)	0.93(+0.09)	1.0(+0.09)	0.99(+0.0)
Affiche modèle	0.76(-0.02)	0.61(+0.09)	0.97(+0.17)	0.75(-0.01)
Article	0.91(+0.21)	0.42(+0.09)	0.86(+0.38)	0.86(+0.2)

TABLE 4.4 – Impact de la détection sur le pipeline utilisant VLAD + SURF

4.2.4 Impact du système de détection sur la reconnaissance d'images

Pour évaluer de la détection et de la segmentation des documents sur les systèmes de reconnaissance d'image, nous avons procédé à une détection sur l'intégralité des images requêtes de notre base de données présentée dans la section 3.2.2. L'objectif étant de comparer les résultats entre les images de requêtes étant passées par un système de détection ou non.

Actuellement, sur les 2 780 images requêtes, 2 233 ont été modifiées grâce à notre système de détection de document, ce qui représente 80%. Nous fournirons les résultats sur ce sous-ensemble de données avec et sans segmentation sur les méthodes ayant obtenu précédemment des résultats encourageants.

Pour le premier cas VLAD + SURF présenté dans le Tableau 4.4, nous observons, de manière générale, une augmentation de l'ensemble des métriques, spécifiquement les mesures de rappel et la précision $K - NN$. Cette augmentation est d'autant plus vraie pour la catégorie des cartes de visites et le cas des factures. Cela témoigne de l'impact de l'arrière-plan lorsqu'il s'agit de groupes de documents à risques ou de petites tailles. Nous notons également une augmentation de la précision pour le cas des cartes de visites, qui est directement en lien avec l'augmentation de la précision $K - NN$. Nous observons également une augmentation de la précision pour les articles ainsi que la précision $K - NN$ pour cette catégorie.

Lors de l'utilisation conjointe des méthodes VLAD et SIFT, l'effet du système de détection apparaît relativement modeste. En effet, bien qu'une amélioration de la précision soit observable pour les cartes

Catégories	Précision	Rappel	$P@40$	Fbeta
Avg	0.78(+0.06)	0.74(+0.04)	0.84(+0.07)	0.78(+0.06)
Carte de visite	0.78(+0.13)	0.71(+0.01)	0.80(+0.01)	0.64(-0.01)
Catalogue	0.92(+0.09)	0.85(+0.11)	0.98(+0.11)	0.91(+0.09)
Facture	0.9(+0.04)	0.84(+0.0)	0.9(+0.02)	0.89(+0.04)
Affiche	0.93(+0.02)	0.87(+0.04)	0.99(+0.07)	0.92(+0.01)
Modèle Affiche	0.42(+0.07)	0.68(+0.04)	0.89(+0.19)	0.43(+0.07)
Article	0.9(+0.06)	0.88(+0.13)	0.9(+0.16)	0.9(+0.06)

TABLE 4.5 – Impact de la détection sur le pipeline utilisant VLAD + SIFT

Catégories	Précision	Rappel	$P@40$	Fbeta
Avg	0.88(+0.03)	0.78(+0.03)	0.93(+0.05)	0.88(+0.04)
Carte de Visite	0.83(+0.38)	0.83(+0.23)	0.9(+0.33)	0.83(+0.37)
Catalogue	0.98(-0.02)	0.87(+0.0)	0.99(+0.02)	0.97(-0.02)
Facture	0.79(-0.01)	0.83(-0.07)	0.7(-0.06)	0.79(-0.02)
Affiche	1.0(+0.0)	0.95(+0.01)	0.99(+0.02)	0.99(-0.01)
Modèle Affiche	0.61(-0.03)	0.47(-0.03)	0.98(+0.0)	0.6(-0.03)
Article	0.99(+0.0)	0.92(+0.06)	0.92(+0.01)	0.98(+0.0)

TABLE 4.6 – Impact de la détection sur le pipeline utilisant GeM + SuperPoint + LightGlue

de visite, les résultats, pour l'ensemble des catégories, semblent analogues à ceux obtenus avec une approche dépourvue de système de détection.

L'augmentation des résultats pour la catégorie des cartes de visites se confirme encore avec la combinaison utilisant GeM + SuperPoint + LightGlue. En effet, l'augmentation de la précision $K-NN$ impacte directement la mesure de rappel et la précision final. Cependant, pour les autres catégories, l'utilisation d'un système de détection influe très peu.

En synthèse, il apparaît que ce type de prétraitement exerce une influence plus ou moins notable sur la précision $K-NN$, conformément à nos suppositions initiales. L'isolation de l'image de son arrière-plan facilite la concentration exclusive des données pertinentes lors de la constitution du vecteur de représentation. Cet élément influe incontestablement sur la sélection de la liste des candidats pour le $K-NN$. L'accroissement de cette précision se répercute, par la suite, sur la mesure du rappel ainsi que sur la précision finale. Néanmoins, pour certaines catégories, comme les modèles d'affiches, il semble que la méthode de reclassement n'atteigne pas la rigueur requise pour identifier adéquatement l'image concernée malgré une augmentation de la précision $K-NN$.

4.3 Identification des documents par sous-images

Suite aux différentes évaluations menées, nous avons identifié un enjeu majeur concernant la recherche et la précision $K-NN$. Cette insuffisance de précision peut entraîner des réponses erronées et, dans le scénario le plus défavorable, un faux positif, notamment, lorsque la base de données renferme des images susceptibles de présenter des similitudes. Les faux positifs surviennent lorsque les méthodes de validation ne parviennent pas à discerner efficacement les nuances, parfois infimes, entre deux images, ou que l'image pertinente n'est pas incluse dans les résultats retournés par la recherche

$K - NN$.

Dans cette optique, nous avons ré-visité l'architecture de notre système de recherche, avec pour ambition d'accroître sa précision, notamment pour les ensembles d'images ayant des caractéristiques communes. Parallèlement, nous accordons une attention particulière à l'efficacité calculatoire du système, le temps de réponse ne devant pas excéder une seconde.

Le principe proposé consiste à abandonner la modalité de recherche globale et adopter une approche semi-globale, par sous-images. Un descripteur est alors associé par la suite à chacune des sous-images. De cette manière, nous espérons obtenir une description d'image plus riche, qui puisse améliorer la précision des résultats.

4.3.1 Présentation du pipeline de recherche

Comme dans les architectures précédentes, notre système opère selon deux modes distincts : le mode "hors ligne" et le mode "en ligne". Ces deux modes poursuivent les objectifs similaires identifiés précédemment : le premier concerne la constitution et l'indexation de la base de données, tandis que le second se focalise sur l'identification à partir d'une image de requête.

4.3.1.1 Traitement hors ligne

Dans le contexte des pipelines traditionnels, la première distinction notable concerne le traitement et l'enregistrement des images de référence. Chaque image de la base de données est subdivisée de manière uniforme en un nombre de N sous-images carrées de dimension $H \times H$.

Chaque sous-image est examinée afin d'identifier l'éventuelle présence d'une occurrence déjà cataloguée dans notre base de données et éviter ainsi les redondances. Notons qu'une sous-image donnée peut être liée à plusieurs images distinctes. Cette approche nous amène à mettre en place un système s'appuyant sur plusieurs bases de données, comme illustré dans la Figure 4.10. La première base est consacrée à une documentation intégrale des images de référence, couvrant leur ensemble visuel et un identifiant unique.

La seconde base se concentre spécifiquement sur les sous-images, englobant des éléments tels que leur identifiant distinctif et leurs vecteurs caractéristiques.

La troisième base a pour mission de consigner la localisation, les dimensions, ainsi que les liens interconnectant les images de référence et leurs sous-images correspondantes. Cela offre la possibilité, lorsqu'une sous-image est identifiée, de retrouver les images mères associées.

L'atout majeur de notre architecture remodelée réside dans sa malléabilité et sa capacité d'adaptation. Elle peut s'articuler soit autour d'une approche descriptive déterministe, soit via l'apprentissage automatique. L'impératif demeure d'obtenir une méthode apte à élaborer une représentation vectorielle distincte de dimension L, pouvant être référencée dans un moteur de recherche s'appuyant sur la méthodologie des k plus proches voisins ($K - NN$). Mentionnons également que la taille des sous-images est normalisée à une résolution de 200x200 pixels.

4.3.1.2 Traitement en ligne

Dans la phase en ligne, la première étape se concentre sur l'identification de sous-images au sein de l'image de requête. Contrairement à la phase hors ligne, où une seule découpe de l'image est réalisée,

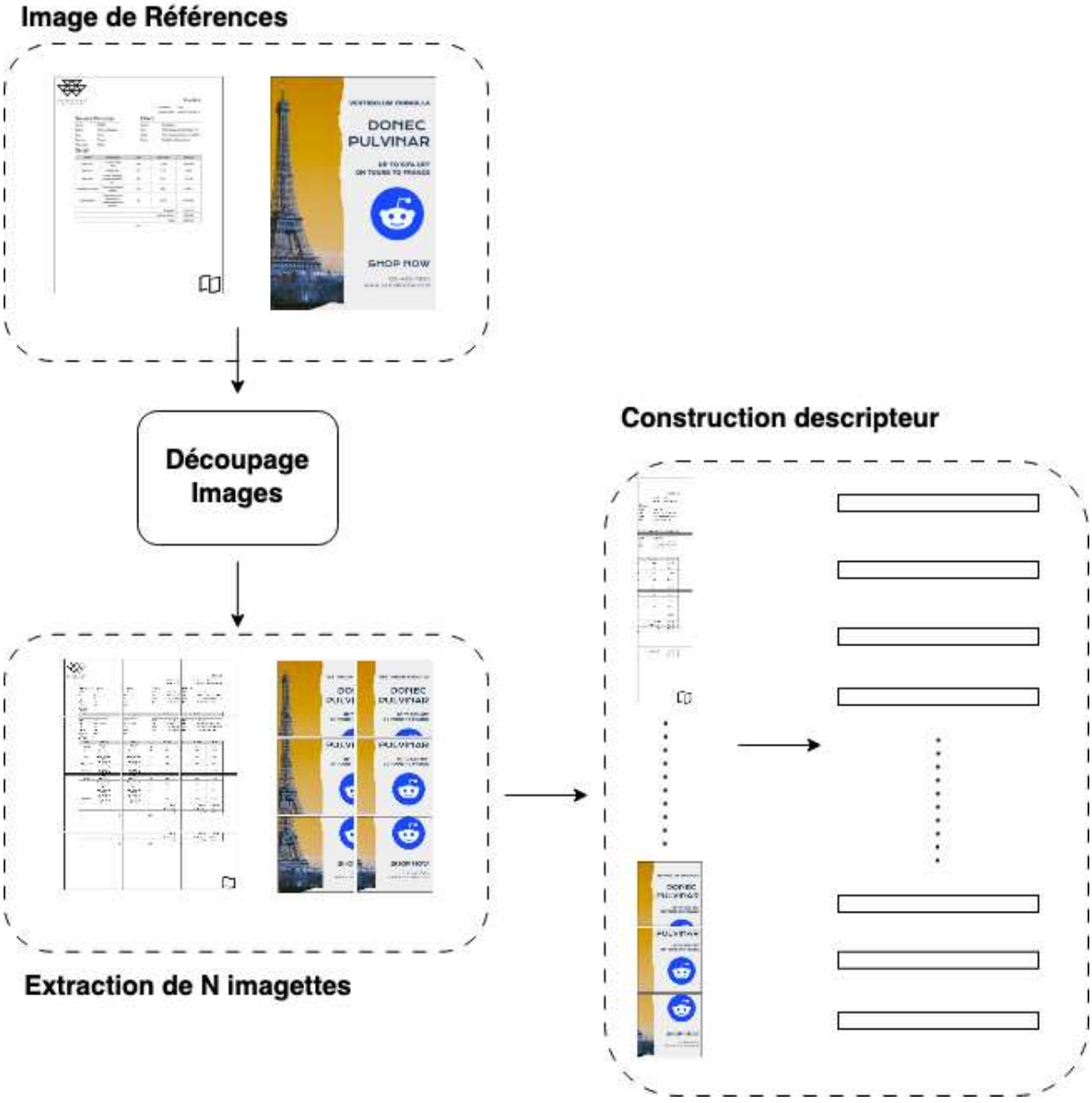


FIGURE 4.8 – Découpage d'images de références puis construction des N représentations vectorielles

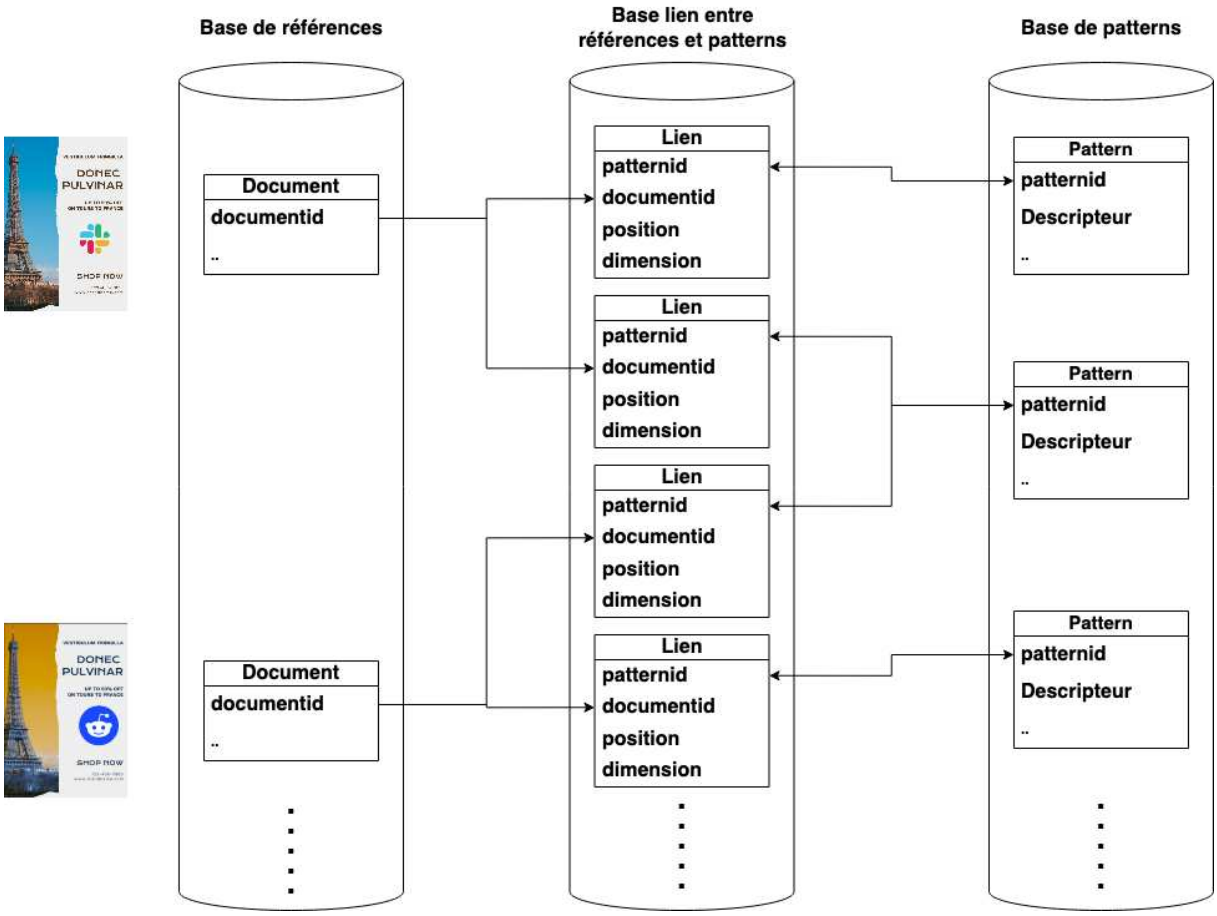


FIGURE 4.9 – Schéma des bases de données nécessaires pour le nouveau pipeline

cette étape effectue de façon hiérarchique par de multiples découpes qui correspondent à de niveaux de détails/échelles croissants (Figure 4.10). À chaque itération, le nombre de fenêtres de recherche dans l'image consultée augmente. L'objectif est de focaliser progressivement l'analyse sur des zones de plus en plus petites, afin de détecter d'éventuelle sous-image pertinentes au sein de cette image de requête.

Chaque sous-image est d'abord normalisée en résolution pour être cohérente avec les sous-images enregistrées dans la base de données. Elle est décrite ensuite par le descripteur visuel considéré. Une recherche par les plus proches voisins ($K - NN$) est alors effectuée, assortie de critères de validation usuels (seuil de distance et ratio de Lowe). Nous examinons, ensuite, s'il y a une transformation perspective compatible entre la sous-image requête et une des K sous-images de la base de données. Si une transformée est identifiée, la première étape s'achève. Sinon, l'analyse se poursuit avec les fenêtres suivantes. Si aucune des fenêtres ne répond aux critères, le système conclut qu'aucune correspondance d'images n'a été trouvée.

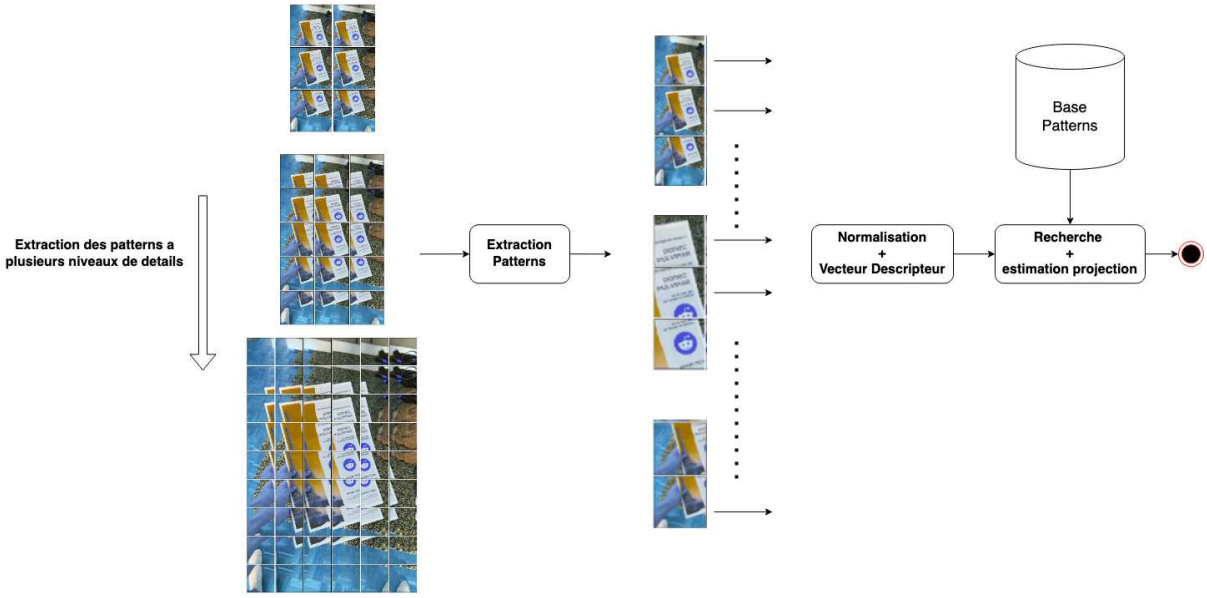


FIGURE 4.10 – Première étape de recherche d'un pattern dans l'image de requête. Le système s'interrompt seulement lorsqu'une sous-image a été identifiée, ou seulement lorsque les N échelles de fenêtres ont été identifiés.

La seconde phase s'opère uniquement à la suite de l'identification d'une sous-image pertinente. L'objectif ici est de déterminer si cette sous-image est spécifique à une seule image de la base ou si elle est commune à plusieurs. Dans l'éventualité où le motif est singulier à une image, les informations d'adjacence définissant la disposition relative entre cette sous-image et ses voisins sont tout simplement récupérées.

Toutefois, si elle appartient à plusieurs images de la base, la disposition est estimée pour chaque image et seules les nouvelles configurations sont ajoutées. Cela implique que pour une sous-image commune à, disons, dix images, la configuration des sous-images adjacentes pourrait demeurer invariable pour chacune d'entre elles. Par conséquent, une unique configuration serait soumise à analyse, comme illustrée dans la Figure 4.11. Cependant, il peut exister des configurations, ou une sous-image peu apparaitre dans plusieurs images, mais a différentes dimensions et positions. Il existera donc, plusieurs configurations de recherche différentes dans ce cas spécifique.

Les séquences d'explorations des différentes configurations sont définies à l'aide d'un arbre couvrant minimal (spanning tree). L'objectif est d'établir des connexions entre les différents nœuds (représentant les barycentres des sous-images contenus dans l'image de référence) tout en optimisant la distance totale parcourue. Le point de départ coïncide avec la sous-image identifiée. À cette fin, une distance L2 est employée pour déterminer les poids relatifs aux différentes arêtes. La localisation des barycentres est approchée dans l'image requise grâce à l'homographie déterminée. La Figure 4.11 illustre un tel parcours. Dans cette représentation, le motif initial identifié est $M1$. Concernant la trajectoire, trois branches distinctes sont identifiables, toutes prenant source en $M1$.

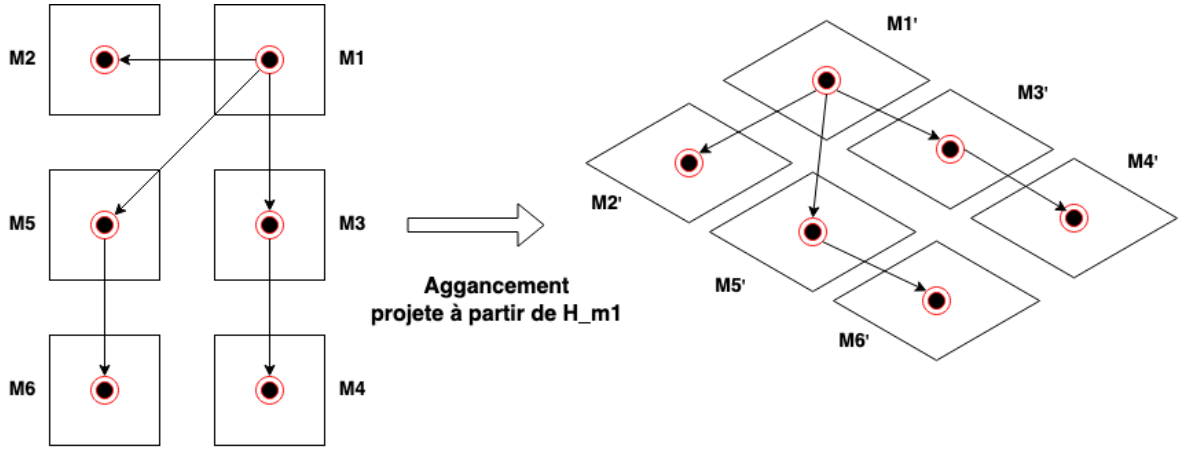


FIGURE 4.11 – Schéma de parcours des nœuds avec leurs projections à partir de H_m1

Dans le cas de la figure 4.11, pour identifier l'image, trois branches doivent être explorées, à savoir : M_1M_2, $M_1M_5M_6$, et $M_1M_3M_4$. Lors de chaque exploration, l'homographie initiale H_{M_1} est appliquée sur la zone de recherche $M'1$. La transformation perspective aboutit à une image rectifiée, désignée $I'1$. Cette image est ensuite analysée en fonction d'une liste de sous-images potentielles. Cette liste est définie pour chaque zone au moment du chargement de la configuration. C'est à partir de la première sous-image identifiées, qu'il est possible de remonter à travers la base de données pour obtenir toutes les images de références potentielles, et en particulier la liste de chaque sous-images potentielles pour chaque zone.

Pour chaque zone, une fois l'image rectifiée obtenue, nous extrayons un descripteur global que le comparera avec l'ensemble des sous-images potentielles de la base de données. Nous estimons donc ainsi une distance L2 (ou autres) pour chaque sous-image potentielles et la zone concernées, et nous définissons un score de vraisemblance selon la relation $sc = \frac{1}{1+\text{Distance}}$, donnant lieu à une valeur entre 0 et 1.

Par la suite, nous cherchons à déterminer l'existence d'une transformation perspective, désignée par H_{PM_1}. Cette transformation correspond à la projection entre la sous-image la plus proche et l'image $I'1$. Si une telle transformation est discernée, H_{M_1} est actualisée selon la relation $H_{M_1} = H_{PM_1}$, avant de poursuivre vers le nœud subséquent.

À titre d'exemple, pour la séquence $M_1M_5M_6$, l'investigation s'oriente ensuite vers la zone M'_5, et

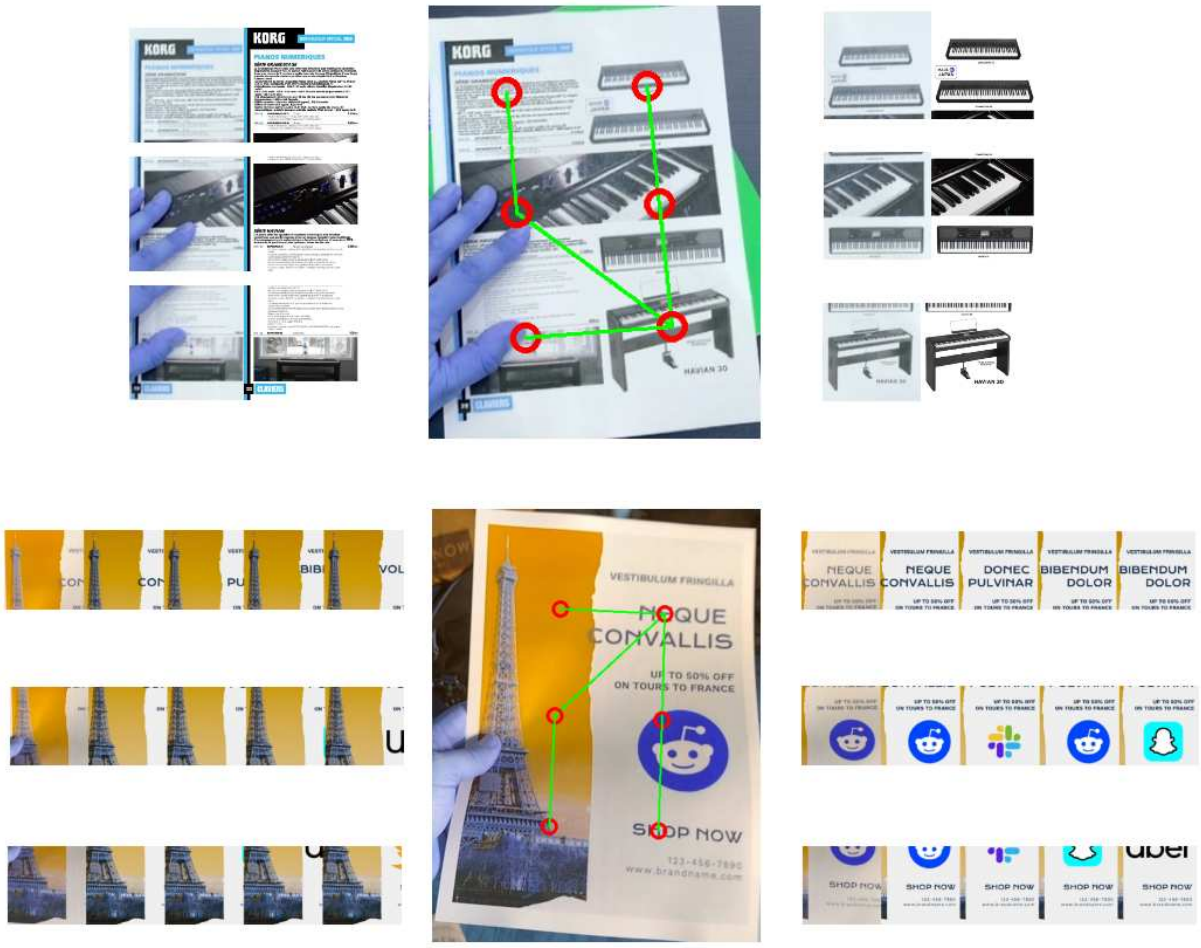


FIGURE 4.12 – Exemple parcours de recherche et motif compare sur chaque zones

la transformation du nœud antérieur est employée pour déduire $I'5$. Dans ce contexte, la transformation perspective de M_1, notée H_{M_1}, est mobilisée. Le processus d'extraction du vecteur caractéristique est de nouveau mis en œuvre, suivi de la quête des sous-images pertinentes. Postérieurement à cette phase, une inspection est conduite pour déceler une transformation entre $I'5$ et la sous-image la plus analogue. En l'absence d'une telle transformation, l'homographie est spécifiée par $H_{M_5} = H_{M_1}$. Autrement, elle est définie par $H_{M_5} = H_{PM_5}$. Cette procédure accentue la robustesse de la requête face à des distorsions potentielles.

Pour conclure, une fois l'intégralité de la configuration passée en revue, le score de chaque image est estimé à partir des scores associés aux différentes zones. Un coefficient est attribué à chaque zone en fonction du nombre de sous-images plausibles. L'ambition est d'accorder une prépondérance aux zones possédant avec un fort potentiel discriminatoire (avec un nombre élevé de motifs potentiels) dans le calcul du score final.

4.4 Évaluation et comparaison

Ce nouveau pipeline offre la possibilité d'adopter différentes méthodes, qu'elles soient descriptives ou basées sur l'apprentissage. Effectivement, il requiert un système d'extraction de descripteur global ainsi qu'un mécanisme d'estimation de transformation perspective.

Lors de nos expérimentations, nous avons utilisé une technique pour l'élaboration de descripteurs globaux : la combinaison VLAD + SIFT, avec différentes dimensions en considération. De surcroît, nous avons également opté pour une méthode s'appuyant sur l'apprentissage. À cette fin, ce modèle (Figure 4.13) repose sur un ResNet50 pour la phase d'encodage, suivi de deux couches de convolutions et d'un système d'attention spatiale et par canal, le tout couronné par un mécanisme d'agrégation des données similaires à GeM et d'une couche entièrement connectée pour la construction d'un vecteur unique de dimension L.

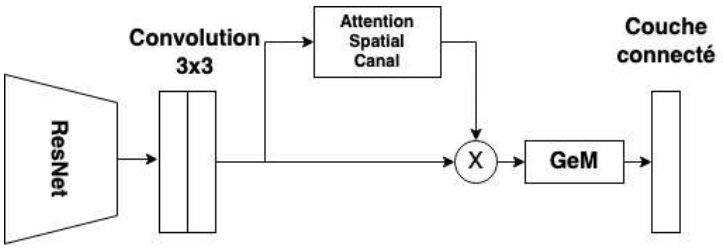


FIGURE 4.13 – Modèle pour la construction des descripteurs globaux

Concernant la formation de ce modèle, nous avons, dans un premier temps, réemployé notre dispositif de création de données synthétiques, cette fois à partir de segments d'images. Ainsi, nous avons pu constituer aisément une base de données composée de 15 000 images de référence, avec 40 images de requêtes pour chacune. Durant la phase d'apprentissage, nous avons adopté les mêmes fonctions d'apprentissage que celles utilisées par DOLG 2. Les images ont été normalisées à une résolution de 224x224, et la formation s'est étendue sur 60 époques.

Pour la détermination de la transformation, notre choix s'est porté sur des descripteurs locaux ORB, avec une estimation de l'homographie par l'intermédiaire de l'algorithme PROSAC. Toutes les images de recherche, qu'il s'agisse des motifs de référence ou des fenêtres de l'image de requête, ont été normalisées à une résolution constante de 200x200, l'ambition étant de réduire les durées de calcul grâce à des images de faible résolution.

Les résultats initiaux, exposés dans le tableau 4.10, révèlent une amélioration notable de la précision pour toutes les méthodes, en comparaison avec les démarches précédemment examinées. En effet, nous observons une précision globale atteignant un plancher de 0.98, voire culminant à 1.0 dans le cas de VLAD pour des vecteurs de dimension 16 384. S'agissant des mesures de rappel, les techniques employant des vecteurs basés sur l'apprentissage profond enregistrent les scores les plus élevés, avec des valeurs de 0,75 et 0,73. Il est à noter que les méthodes axées sur l'apprentissage semblent offrir le meilleur équilibre entre précision, rappel et dimension des vecteurs.

Il convient, également, de souligner une limitation de ce nouveau protocole : sa durée d'exécution. En effet, nos observations indiquent qu'en moyenne, le processus requiert approximativement 1,3 seconde, que nous utilisons des méthodes basées sur l'apprentissage ou sur la description. Bien que cette durée demeure relativement acceptable, elle s'avère particulièrement affectée lorsque le système doit explorer

Méthodes	Dimension	Précision	Rappel	Fbeta	Temps
VLAD SIFT	16 384	1.0	0.62	0.97	1.38
	8192	0.99	0.59	0.95	1.32
	4096	0.99	0.56	0.95	1.33
	2048	0.98	0.52	0.93	1.30
ResNet50 + Attention + GeM	2048	0.98	0.75	0.96	1.25
	1024	0.98	0.73	0.96	1.25

TABLE 4.7 – Évaluation du nouveau pipeline avec différentes méthodes de construction de vecteurs

un grand nombre de fenêtres afin d'identifier une première sous-image. Ensuite, une fois cette première image identifiée, la deuxième phase fonctionnement très rapidement (inférieur à 0.5 sec).

En synthèse, il apparaît que, à dimensionnalité équivalente, VLAD enregistre des performances inférieures par rapport à la méthode basée sur l'apprentissage. Par ailleurs, l'exploitation de vecteurs d'une dimensionnalité supérieure à 2048 s'avère inenvisageable, compte tenu des contraintes mémorielles conséquentes qu'une telle approche induirait.

En examinant de plus près les résultats pour chaque catégorie en utilisant les méthodologies basées sur l'apprentissage profond, une tendance se dégage : la précision diminue, comme attendu, pour les catégories jugées plus complexes, notamment les factures, les cartes de visite et les modèles d'affiches. Néanmoins, il est à noter que la précision demeure relativement élevée pour ces deux dernières catégories, affichant des scores de 0,98 pour les vecteurs de dimension $L = 2048$ et des scores de 0,96 et 0,99 pour ceux de dimension $L = 1024$. La seule catégorie qui présente une précision notablement plus faible est celle des factures, avec des scores de 0,91 et 0,88 respectivement. Ces résultats suggèrent une moindre capacité des vecteurs de représentation à distinguer des images présentant des éléments textuels variés, mais disposés selon des mises en page identiques. La résolution utilisée peut également impacter les résultats ainsi que la surface que les images représentent dans l'image de référence.

Dans un second temps, il est pertinent d'évaluer l'impact de notre système de détection sur ce pipeline de recherche innovant. Les données présentées dans le tableau 4.9 indiquent que l'introduction de notre système de détection en amont n'entraîne qu'une infime variation des résultats, que ce soit en termes de précision ou de rappel. Cette quasi-stabilité s'explique principalement par la méthodologie adoptée par ce nouveau pipeline. Notamment, le rappel demeure constant, car la première étape implique une recherche par fenêtrage multi-niveaux. Ainsi, que le système de détection soit activé ou non, les fenêtres analysent les mêmes régions d'intérêt. Concernant la précision, cette métrique reste également invariante du fait que la méthode de validation examine exhaustivement toutes les configurations possibles en fonction de la première sous-image identifie.

Il convient toutefois de souligner un impact positif notable sur le temps d'exécution. Cette réduction du temps d'exécution est cohérente avec les observations relatives au rappel. Étant donné que le processus débute par une recherche par fenêtrage, si le système de détection en amont réussit à cerner l'image cible, un nombre réduit de fenêtres sera alors exploré, diminuant ainsi le temps d'exécution.

Une seconde utilisation du système de détection pourrait alors être l'utilisation comme une carte de chaleur permettant de déterminer et de focaliser les zones ou les fenêtres de recherche seront construites.

Catégories	*Précision*	*Rappel*	*Fbeta*	*Temps*
avg	0.98	0.73	0.96	1.25
Carte de Visite	0.98	0.49	0.93	2.0
Catalogue	0.99	0.86	0.98	0.98
Facture	0.91	0.73	0.9	1.88
Affiche	1.00	0.89	0.99	0.97
Modèle Affiche	0.98	0.66	0.95	0.85
Article	0.99	0.75	0.97	1.30

(a) ResNet50 + Attention + GeM Dimension $L = 2048$

Précision	*Rappel*	*Fbeta*	*Temps*
0.98	0.75	0.96	1.25
0.96	0.49	0.91	2.18
1.00	0.89	0.99	0.86
0.88	0.75	0.87	2.12
1.00	0.91	0.99	0.94
0.99	0.67	0.96	0.79
0.99	0.79	0.98	1.24

(b) ResNet50 + Attention + GeM Dimension $L = 1024$

TABLE 4.8 – Tableau détaillant les résultats pour chaque categories

Catégories	Précision	Rappel	Fbeta	Temps
avg	0.98 (+0.00)	0.75 (+0.00)	0.96 (+0.00)	1.14 (-0.12)
Carte de Visite	0.96 (+0.00)	0.59 (+0.10)	0.93 (+0.02)	1.43 (-0.76)
Catalogue	0.99 (-0.01)	0.86 (-0.03)	0.98 (-0.01)	0.95 (+0.08)
Facture	0.91 (+0.03)	0.77 (+0.02)	0.90 (+0.03)	1.89 (-0.24)
Affiche	1.00 (+0.00)	0.90 (-0.01)	0.99 (+0.00)	0.81 (-0.13)
Modèle Affiche	0.99 (+0.00)	0.68 (+0.01)	0.96 (+0.00)	0.74 (-0.06)
Article	0.99 (+0.00)	0.75 (-0.04)	0.97 (-0.01)	1.42 (+0.17)

TABLE 4.9 – Évaluation avec le système de détection de document

Méthode	sec	$P@40$	Précision	Rappel	FBeta
VLAD + SIFT	0.15	0.81	0.71	0.75	0.71
VLAD + SURF	0.28	0.72	0.88	0.57	0.85
ResNet50 + GeM + SuperPoint + Glue	1.8	0.86	0.86	0.72	0.85
Vuforia	0.35	-	0.89	0.77	0.86
VLAD 128	1.38	-	1.0	0.62	0.97
ResNet50 + Attention + GeM 2048	1.25	-	0.98	0.75	0.96

TABLE 4.10 – Comparaison des différentes méthodes

Comparativement aux méthodes ayant affichées les performances les plus probantes, la première observation concerne indubitablement la précision globale. En effet, même en instaurant des seuils d'une exigence exceptionnelle, les approches antérieures ne parviennent pas à atteindre une telle précision, malgré une baisse significative du rappel. Or, ce nouveau protocole permet simultanément d'affiner l'identification en minimisant les erreurs de confusion et de maintenir un niveau de rappel satisfaisant, en particulier lorsque nous employons un descripteur basé sur l'apprentissage.

Par ailleurs, nous avons évalué la solution proposée par Vuforia. Bien que cette dernière demeure une "boîte noire" nous privant de toute connaissance sur sa méthodologie, les résultats obtenus sur notre base de données se révèlent inférieurs. Cette constatation renforce l'attractivité de notre approche pour une utilisation en production de ce système de recherche.

4.5 Bilan

En conclusion, l'orientation de nos recherches, centrée sur des applications industrielles, a révélé un potentiel notable. Concernant la précision, nos résultats se rapprochent considérablement de l'objectif, avec un taux avoisinant les 100%. Cependant, le rappel reste en deçà de nos aspirations, s'établissant à 0,75. Il est concevable d'accroître ce score en augmentant par exemple le nombre de fenêtres de recherche ou de modifier les paramètres de création des fenêtres, bien que cette augmentation pourrait affecter le temps d'exécution et l'espace mémoire nécessaire.

De plus, cette stratégie de recherche pourrait induire une consommation mémorielle considérable. Malgré la réduction de la dimensionnalité des vecteurs caractéristiques, leur quantité a augmenté. À titre illustratif, pour une base de 100 000 images, avec l'extraction d'environ six sous-images par image, cela équivaudrait à 600 000 sous-images. Avec des vecteurs de dimension $L = 16384$, la consommation

mémorielle s'élèverait à 39,3 Go. Il est donc manifeste que l'usage de VLAD SIFT avec de tels vecteurs n'est pas envisageable actuellement. Cependant, avec des vecteurs résultant de méthodes d'apprentissage de dimension $L = 1024$, la consommation mémorielle serait de 2,5 Go, ce qui semble à la fois viable et prometteur.

Une première implémentation de notre solution, basée sur cette stratégie, est en cours d'évaluation et devrait être prochainement déployée en production. Pour les étapes ultérieures, diverses améliorations se présentent. La première concerne l'optimisation algorithmique, avec pour ambition de réduire les temps de calcul à moins d'une seconde.

En outre, des modifications pourraient être envisagées au niveau du système de création des vecteurs de représentation, dans le but de les condenser davantage. Les vecteurs issus des méthodes d'apprentissage ont démontré leur efficacité, offrant d'excellentes performances avec des dimensions réduites. Nos évaluations préliminaires, calquées sur un modèle similaire à GeM, s'avèrent encourageantes. Il serait toutefois pertinent d'explorer des alternatives pour davantage réduire la taille des vecteurs caractéristiques, minimisant ainsi leur empreinte mémorielle, tout en améliorant leur performance sur les images comportant des éléments textuels.

Des initiatives visant à substituer la méthode d'estimation de la matrice d'homographie ont également été initiées. Dans ce cadre, le recours au transformeur spatial [JSZ+15], a été privilégié. Toutefois, les résultats préliminaires ne montrent pas d'amélioration significative face à une augmentation du coût calculatoire. Malgré cela, il semble judicieux de maintenir cet axe de recherche, étant donné que de telles méthodes ont prouvé leur efficacité pour corriger également les déformations documentaires, comme démontré par [Xue+22].

FIGURE 4.14 – Exemples de résultats provenant de notre base de données.

FIGURE 4.15 – Exemples de résultats provenant de notre base de données.

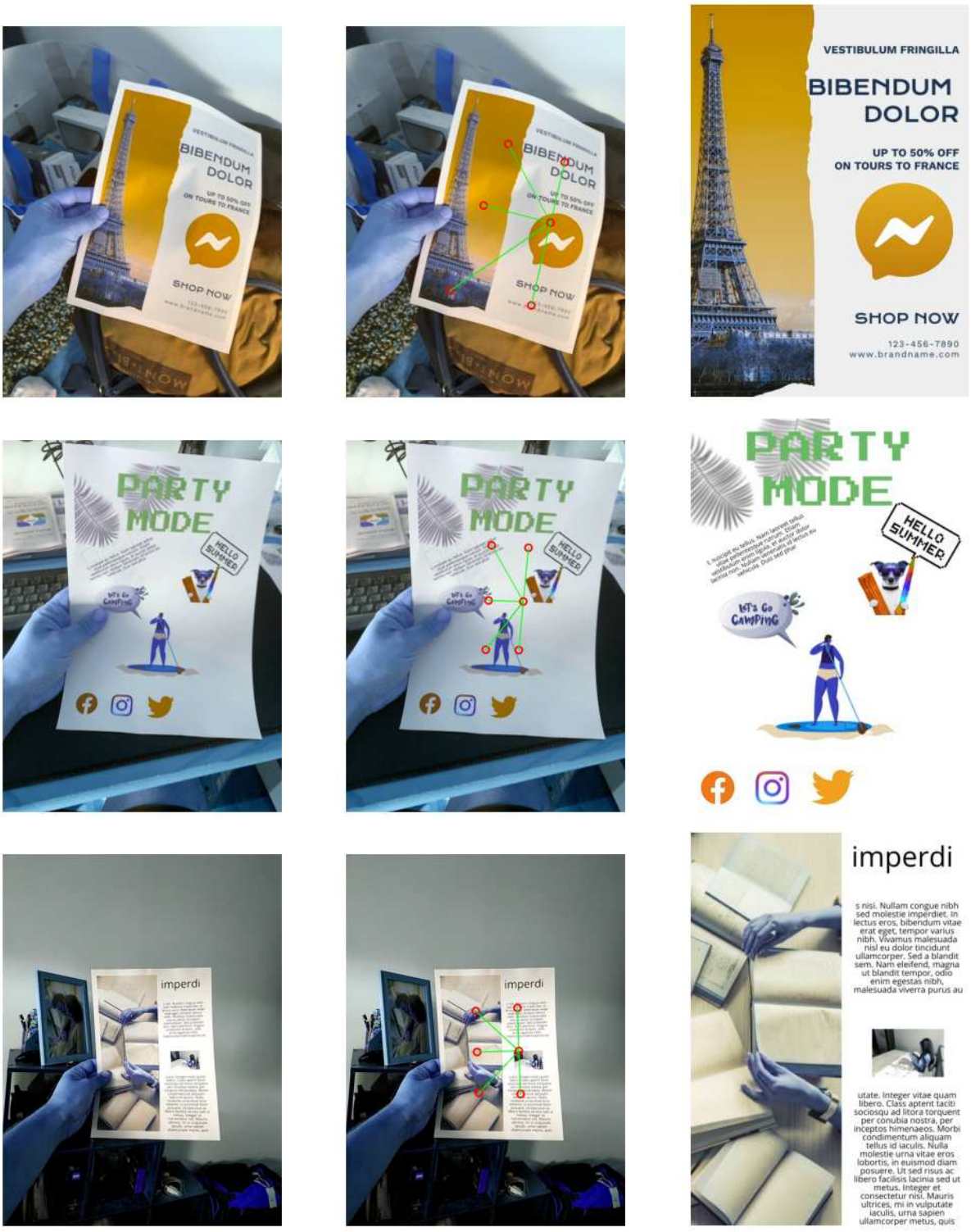


FIGURE 4.16 – Exemples de résultats provenant de notre base de données.

Chapitre 5

Développement d'un moteur Web de suivi d'images

Ce chapitre détaille le développement d'une librairie en C++ convertie en WebAssembly, conçue spécifiquement pour le suivi d'images en temps réel. Cette technologie avancée tire profit de la rapidité et de la portabilité du WebAssembly, rendant possible un suivi d'image performant directement dans les navigateurs web. La librairie est actuellement intégrée dans une application web de réalité augmentée basée sur des marqueurs, illustrant son efficacité et son adaptabilité à des scénarios d'utilisation en direct. Ce développement ouvre des portes à de nouvelles perspectives en matière d'interactions en réalité augmentée sur le web.

5.1 Introduction

Dans ce chapitre, nous allons aborder différents travaux, davantage orientés sur l'industrie, autour de la conception d'un moteur de suivi d'images pour les applications Web. En effet, la société ARGO propose et commercialise une plateforme dotée d'un éditeur permettant à tous les clients de déposer de nouveaux marqueurs (par exemple la couverture d'un magazine) et créer une expérience de réalité augmentée en y ajoutant divers contenus. Il s'agit d'une expérience à base d'un marqueur 2D planaire.

Historiquement, la plate-forme ARGO était une application native : il était nécessaire pour chaque utilisateur de télécharger et d'installer l'application sur son appareil mobile. La diversité des marques et de leurs modèles d'appareils mobiles posait de nombreuses problématiques. En effet, maintenir une application mobile compatible avec différents systèmes d'exploitation (OS), ainsi qu'avec plusieurs modèles d'appareils nécessitait la maîtrise de plusieurs outils et langages de programmation.

Contrairement aux applications natives, les applications Web présentent de multiples avantages. Elles simplifient le développement multiplateforme en réduisant le nombre d'outils et de langages de programmation à maîtriser. En outre, elles n'exigent pas d'installation préalable. Pour l'utilisateur, un QR code suffit pour accéder à l'URL de l'application et l'activer. C'est pour cette raison que les applications Web deviennent un véritable enjeu [Qia+19], notamment dans le domaine de la réalité augmentée.

Dans ce sens, nous retrouvons différentes bibliothèques écrites uniquement en JavaScript permettant de créer des expériences en réalitée augmentée sur le Web, notamment JS-ArUco (un portage en JavaScript de l'ArUco), JSARToolkit (basé sur l'ARToolKit original), JSARToolKit5 (un portage emscripten de l'ARToolKit) et AR.js basée sur Three.js et JSARToolKit5. Ces solutions permettent d'obtenir de bonnes performances (jusqu'à soixante images par seconde dans certains cas). Cependant, elles ne permettent que la détection et le suivi de marqueurs fiduciaires, ce qui nécessite des méthodes simples et peu complexes sur le plan algorithmique.

Dans notre cas, les marqueurs sont des images qui comportent des contenus hétérogènes (graphiques et textuels) et qui peuvent ne pas être répartis uniformément dans le marqueur. Les méthodes de détection et de suivi de ce type de marqueur nécessitent l'utilisation de caractéristiques descriptives qui sont beaucoup plus coûteuses et complexes à mettre en place. C'est l'un des principaux problèmes dans le développement de solutions Web basées sur des approches par description. En effet, les environnements et langages web ont des performances bien inférieures aux applications natives. Ces limitations sont, principalement, causées par les contraintes imposées aux ressources du processeur ou du GPU. Il existe également des problématiques liées aux langages interprétés tels que JavaScript, qui sont bien moins performants que les langages compilés.

De plus en plus d'entreprises et de concurrents, tels que les sociétés 8Th WALL et ZAPPAR, ont développé leurs propres solutions propriétaires permettant la création d'expériences de réalité augmentée dans un navigateur Web. C'est en réponse à la demande croissante des clients et à la multiplication des concurrents sur le marché proposant ce type d'alternative aux applications natives, que nous avons développé notre propre solution utilisable dans un navigateur Web qui s'avère un enjeu stratégique critique pour la société Argo.

5.2 Les technologies Web

Au cours des dernières années, nous avons pu observer l'avancée de certaines technologies permettant de répondre aux exigences des applications Web de réalité augmentée, qui nécessitent des calculs intensifs. Ces avancées permettent d'obtenir des gains de performances et d'offrir l'opportunité de créer des expériences en temps réel.

Nous retrouvons tout d'abord WebRTC (Web Real Time Communication) qui est une technologie qui permet l'ouverture de canaux de communication en temps réel pour les navigateurs Web. Dans le cadre des applications Web de réalité augmentée, cela pourrait permettre la transmission en temps réel du flux vidéo de la caméra et le déport des processus de traitement d'images vers une instance serveur. Cependant, cette approche se révèle très consommatrice en bande passante et n'est pas raisonnablement applicable dans le domaine industriel, sans compter la charge supplémentaire qu'elle entraînerait sur les serveurs.

Ensuite, WebAssembly [Møl18], ou wasm, est un format d'instruction binaire pour une machine virtuelle. Ce format de fichier binaire permet d'être chargé en tant que module directement par du JavaScript et permet de fonctionner dans un navigateur à une vitesse quasi native du CPU. L'avantage, dans notre cas, est de développer nos fonctions et notre système de détection et de suivi d'images directement en C, C++, Go ou Rust, puis de les utiliser en tant que module depuis notre application web. Il existe ainsi une version de OpenCV en WebAssembly, Opencv.js [Tah+18; Tah+17], qui permet d'accéder à certaines fonctions.

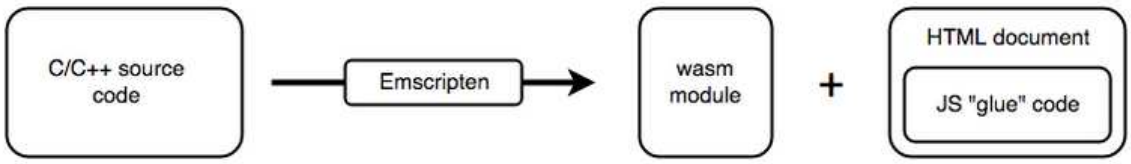


FIGURE 5.1 – Schéma de passage d'un code C++ à sa compilation puis à son utilisation d'une application Web. Le compilateur utilisé est Emscripten

Il existe également les WebWorkers qui permettent l'introduction du multi-threading pour le langage JavaScript. Cela permet d'exécuter différents processus en parallèle, tels que le rendu ou la détection et le suivi d'images dans le cas des applications Web de réalité augmentée. Cette méthode est essentielle et permet d'améliorer l'expérience des utilisateurs et de profiter simplement de l'augmentation du nombre de cœurs des processeurs pour les appareils mobiles.

Enfin le WebGL permet l'utilisation du GPU pour accélérer les rendus dans les applications web. Étant donné que les processus de rendu et de traitement d'images sont très gourmands en ressources informatiques, cette bibliothèque utilisable depuis un langage tel que JavaScript via une API, permet de rendre les rendus et les expériences de réalité augmentée plus fluides.

Ces différentes technologies sont toujours en développement et évoluent rapidement. Elles fournissent une base d'outils indispensables pour le développement et la construction d'applications Web de réalité augmentée. Ainsi, au moins quatre méthodes possibles peuvent être identifiées :

1. Utilisation du langage JavaScript : il est possible d'utiliser différents Web Workers, chacun spécialisé dans une partie du traitement de l'image.

2. Passer le pointeur mémoire du canvas au Wasm : le Wasm traitera alors le flux vidéo et ajoutera lui-même la couche 3D. Cette solution semble la plus performante, elle présente cependant un inconvénient qui a semblé bloquant pour l'affichage. Le moteur 3D doit être codé et compilé en C++.
3. Effectuer le traitement complet en WebGL (via des shaders) en utilisant la carte graphique : ce traitement est optimal pour les opérations locales telles que le passage en niveaux de gris et le flou. Cependant, le développement est complexe pour nos tests en mode itératif.
4. Les données sont transmises du canvas au Wasm, qui enverra à son tour la position de la référence, à condition que cette dernière soit détectée. L'affichage des augmentations s'effectue alors en dehors du Wasm, en utilisant un moteur 3D JavaScript (Three.js). L'avantage substantiel est qu'il est possible de conserver toute la flexibilité du Web pour personnaliser les affichages et d'utiliser une librairie de rendu 3D populaire et maintenue.

La quatrième solution est apparue comme la plus adaptée aux besoins des clients de la société ARGO, dont la majorité a des exigences sur mesure. Cette solution permet de proposer très rapidement des interactions, des jeux et des effets graphiques avancés (shaders WebGL) sans devoir modifier le moteur de suivi d'images.

5.3 Architecture logicielle

Afin de garantir une flexibilité optimale de l'application Web, il nous est apparu indispensable de moduler l'architecture proposée en trois blocs distincts, permettant ainsi une indépendance dans le développement de chacun. Plkus précisément, l'architecture retenue inclut les blocs suivants :

— Une/des webapp(s) pour mettre en place une UX pour l'utilisateur final. Elle peut être développée sans connaissances particulières et sans modifications des deux couches suivantes. Dans le cas industriel, cette couche permet d'adapter un produit rapidement aux couleurs et envies d'un client.
— Une librairie Typescript qui propose toutes les fonctionnalités nécessaires pour créer des expériences 3D à partir de la plateforme de la société ARGO. Cette librairie permet également d'effectuer les requêtes nécessaires aux serveurs de reconnaissance d'images ou au chargement des contenus adéquats.
— Une librairie wasm, qui gère la recherche et le suivi d'une référence dans un flux vidéo et retourne la position/rotation. Cette couche est la plus critique puisqu'elle nécessite d'implémenter des algorithmes de traitement d'images très coûteux en calcul et donc une optimisation particulière.

FIGURE 5.2 – Schéma de l'architecture générale

Le principal enjeu a donc été le développement de la dernière couche, qui correspond à la librairie C++ et dont les objectifs et les contraintes sont les suivants :

1. Temps réel : Obtention d'un nombre d'images par seconde le plus haut possible, y compris sur des modèles d'appareils mobiles bas de gamme.
2. Poids de la librairie : Le poids du fichier wasm à charger à chaque utilisation de l'application web doit être le plus compact possible. Il est, ainsi, nécessaire de limiter, au maximum, l'utilisation de librairies annexes.
3. Précision : Obtenir le système le plus précis possible et limiter les différences entre les expériences sur les applications native et Web.

5.4 Fonctionnement du moteur de suivi d'images

Les applications de réalité augmentée avec marqueurs visent à superposer des informations dont la position et l'orientation sont alignées avec celles du marqueur de référence. Un des défis majeurs est l'estimation en temps réel de la position et de l'orientation du marqueur afin d'assurer une cohérence dans le suivi de ce dernier lorsqu'il est en déplacement. Pour réaliser cela, il est impératif d'estimer la position de la caméra par rapport au marqueur de référence. Grâce à l'évaluation de la position pour chaque image d'une séquence vidéo, un moteur graphique 3D est actualisé.

Puisque les applications de réalité augmentée nécessitent une caméra, notre analyse s'appuie sur le modèle de caméra pinhole. Dans ce contexte, la caméra projette les points d'un espace tridimensionnel (x, y, z) sur un pixel bidimensionnel dans le plan de l'image (u, v, k), où k est un paramètre d'échelle pour les coordonnées homogènes. Mathématiquement, cette transformation est exprimée par :

$$\begin{bmatrix} ku \\ kv \\ k \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_{xy} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} r_{12} r_{13} t_1 \\ r_{21} r_{22} r_{23} t_2 \\ r_{31} r_{32} r_{33} t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5.1}$$

Les coefficients c_x et c_y correspondent à l'origine des coordonnées de l'image, tandis que f_x et f_y dénotent les focales de la caméra, régulant ainsi l'échelle. La première matrice, construite à partir de ces éléments, est qualifiée de matrice intrinsèque, elle est caractéristique à chaque caméra et ne dépend pas de la scène.

La matrice extrinsèque renferme la transformation des coordonnées du monde réel vers le système de référence de la caméra. Elle est formée de quatre vecteurs : trois vecteurs de rotation R_1, R_2 et R_3, ainsi qu'un vecteur de translation t.

Dans le cadre de notre étude, l'objectif est d'estimer la transformation entre deux images planes. Il est relativement aisé d'estimer la matrice d'homographie. À partir de cette dernière, l'ambition est alors d'évaluer la matrice de projection ou matrice extrinsèque afin de pouvoir estimer la position de la camera dans un envrionnement 3D.

$$\begin{aligned}\begin{bmatrix} ku \\ kv \\ k \end{bmatrix} &= A[R_1 R_2 R_3 t] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \\ &= A[R_1 R_2 t] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\end{aligned} \tag{5.2}$$

Dans l'équation ci-dessus, en partant du principe que des projections en 2D sont étudiées, la coordonnée z peut être ignorée, ainsi que la colonne R_3. La relation suivante peut donc être définie :

$$A[R_1 R_2 t] = H \tag{5.3}$$

$$[G_1 G_2 G_3] = A^{-1} H \tag{5.4}$$

Il est alors possible d'associer $G_1 = R_1$ à $G_2 = R_2$ et $G_3 = t$.. Puisque la matrice $[R_1 R_2 R_3 t]$ représente une transformation homogène, $[R_1 R_2 R_3]$ est orthonormée. R_3, pourrait être évalué comme le produit vectoriel de R_1 et R_2, néanmoins, R_1 et R_2 sont simplement des approximations. Il est nécessaire de déduire une base orthogonale $R'1R'2$ à partir de $R_1 R_2$. Pour estimer R_3, nous calculons donc le produit vectoriel $R'1 \times R'2$, permettant alors d'obtenir la matrice extrinsèque définitive.

5.4.1 Détection du marqueur

L'estimation de la matrice extrinsèque, facilitée par l'homographie, permet de superposer un contenu 3D sur le marqueur de référence. La méthode la plus efficiente et rapide pour la détection d'une image et le calcul d'une homographie repose sur l'utilisation de descripteurs locaux. Bien qu'il existe diverses méthodes de construction de ces descripteurs, nous avons opté pour le descripteur ORB [Rub+11] présenté dans le premier chapitre.

Un des principaux atouts de l'ORB [Rub+11] réside dans la nature de ses vecteurs, principalement binaires, diminuant ainsi l'empreinte mémoire par rapport à des descripteurs à valeurs flottantes. Pour l'appariement des descripteurs entre deux images, nous employons une méthode de force brute (recherche exhaustive) en utilisant la distance de Hamming.

Quant à la détection des points clés, ORB emploie l'algorithme FAST [RD06], appliqué à différentes résolutions de l'image source, afin d'isoler efficacement des points d'intérêt à diverses échelles. Dans notre contexte, l'efficacité computationnelle est essentielle. Par conséquent, le détecteur YAPE [LNK18], reconnu pour sa rapidité, a été retenu, étant mieux adapté pour les dispositifs mobiles et embarqués.

5.4.2 Suivi du marqueur

Toutefois, la détection du marqueur est computationnellement exigeante. Il est par conséquent ardu de s'appuyer exclusivement sur ORB pour estimer la position du marqueur dans une séquence vidéo en temps réel. De surcroît, ces descripteurs ont du mal à gérer les perspectives extrêmes, résultant en la perte du marqueur dans la séquence, entravant ainsi l'expérience utilisateur.

FIGURE 5.3 – Suivis de points d'intérêt avec Lucas-Kanade

Afin de pallier ces défis, les méthodes de flux optiques sont couramment utilisées pour le suivi d'objets. Nous distinguons deux catégories : le ”sparse”, qui évalue le mouvement de points spécifiques à travers les frames, et le ”dense”, qui analyse le flux sur l'intégralité de l'image.

Dans le contexte d'applications Web, la priorité est de minimiser les calculs. Ainsi, les approches ”sparse” sont préférées. L'algorithme de Lucas-Kanade [REF] est adopté pour sa célérité, sa simplicité et son efficacité permettant ainsi de suivre un ensemble de points entre deux images (comme illustré sur la Figure 5.3).

Néanmoins, les méthodes basées sur le calcul de flux optiques sont sujettes à une dérive progressive. Cela implique que les points peuvent s'éloigner graduellement, compromettant le calcul de l'homographie. Ce biais peut sérieusement altérer le rendu. Il est alors impératif de concevoir un système combinant les atouts des techniques d'appariement et de flux optique.

5.5 Implantation

Dans cette section, nous procéderons à une exploration méthodique de plusieurs aspects du projet. Tout d'abord, nous nous pencherons sur la constitution de la base de données élaborée spécifiquement pour l'évaluation du moteur de suivi d'images. Ensuite, une présentation exhaustive des fonctions conçues pour la bibliothèque sera mise en avant. Par la suite, nous détaillerons de manière intégrale le pipeline de suivi d'images. En conclusion, une analyse des performances de ce pipeline sur divers dispositifs et

navigateurs Web sera conduite.

5.5.1 Constitution de la base de données expérimentale

Pour procéder à une évaluation rigoureuse du moteur de suivi d'images, la mise en place d'une base de données spécifique est indispensable. Cette base a pour finalité de refléter fidèlement les cas d'usage en contexte industriel. Dans cette optique, nous avons sélectionné cinq documents présentant diverses configurations de mise en page. Pour chacun de ces documents, trois séquences vidéo ont été capturées.

La première séquence se caractérise par des mouvements de caméra strictement translationnels. La seconde intègre simultanément des variations de position et d'orientation de la caméra. La troisième séquence, quant à elle, implique des modifications de position et d'orientation du document lui-même, induites par une intervention manuelle. Ce dernier scénario, en raison de sa complexité et de sa variabilité, représente un véritable défi pour le moteur de suivi d'images.

Chaque séquence se compose d'une série de 180 images environ. L'évaluation de la précision du suivi s'appuiera sur la distance L2 moyenne entre les positions projetées des quatre sommets du document. L'ambition sous-jacente à cette démarche est d'allier précision et rapidité pour assurer une performance optimale du système.

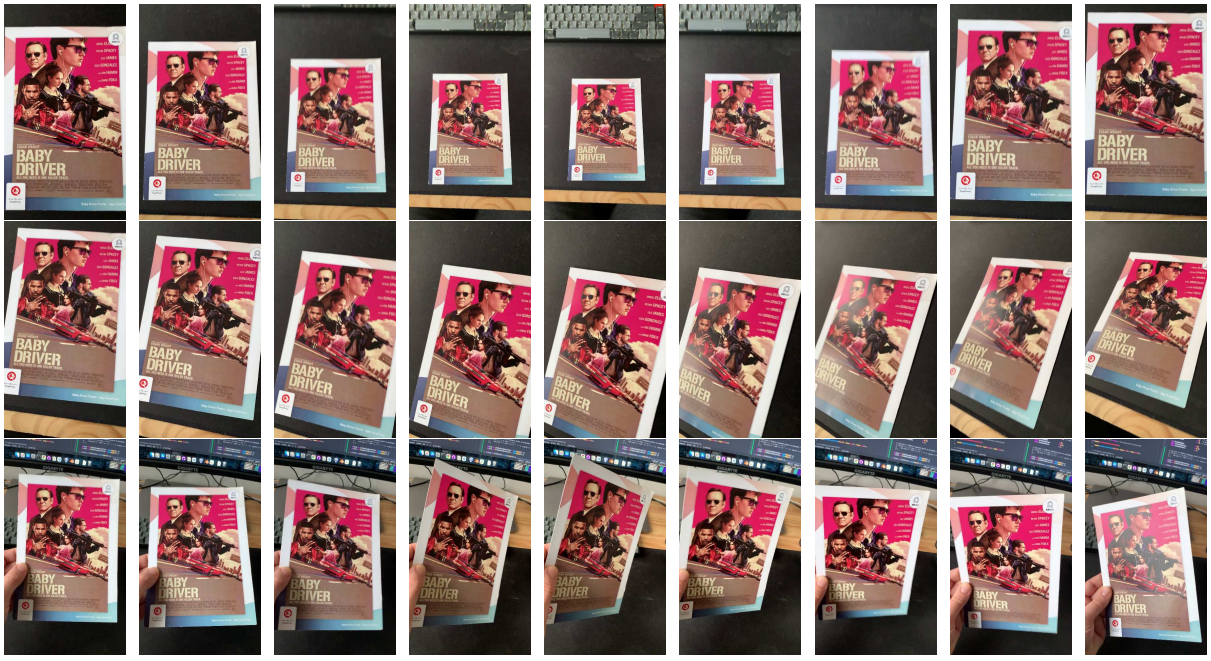


FIGURE 5.4 – Voici des exemples de séquences d'images provenant des trois types de vidéos différentes. La première colonne présente une séquence illustrant des mouvements de translation de la caméra. La deuxième colonne montre des mouvements de rotation autour du marqueur. La troisième colonne présente un cas plus complexe et représentatif des problématiques industrielles, avec un marqueur tenu par un utilisateur.

5.5.2 Librairie de traitement d'image

Plusieurs bibliothèques C++ existent pour le traitement d'images. Parmi elles, OpenCV est la plus reconnue et offre des fonctionnalités robustes pour développer des systèmes de suivi d'images destinés à des applications de réalité augmentée. Cependant, notre analyse a révélé deux principales limitations lors de l'utilisation de cette bibliothèque pour des applications Web.

Premièrement, la dimension de la bibliothèque compilée est préoccupante. Notre prototype initial basé sur OpenCV génère un fichier wasm d'environ 5 Mo une fois compilé, dimension non optimale pour les applications Web, étant donné que les ressources doivent être téléchargées depuis un serveur à chaque démarrage de l'application.

Deuxièmement, il y a la question de la performance. Ces bibliothèques tendent à avoir une consommation de mémoire élevée, notamment en raison de l'allocation de divers objets intermédiaires. Cette caractéristique limite les performances et la réactivité, en particulier pour les appareils de milieu et bas de gamme, qui représentent une grande partie du parc d'appareils actuels. Ainsi, avec le premier prototype base sur OpenCV, nous avons enregistré un taux de traitement moyen de 10 à 15 images par seconde sur des dispositifs de gamme intermédiaire.

Ainsi, nous avons entrepris de développer une bibliothèque personnalisée de traitement d'images. Notre objectif était d'optimiser la gestion de la mémoire et de minimiser les calculs superflus. Pour cela, nous avons choisi d'utiliser le langage C++ et de restreindre l'usage de bibliothèques tierces. Nous avons établi les structures fondamentales de notre bibliothèque, notamment les structures "Images" et "Points", et introduit des fonctions primaires de traitement d'images, telles que :

— Conversion d'un buffer RGB en niveaux de gris
— Application d'un filtre paramétrable (Gaussien, Laplacien, etc.),
— Construction d'une pyramide d'images à échelles variables.

Nos fonctions ont des performances comparables à celles d'OpenCV, mais avec une empreinte mémoire considérablement réduite. Chaque fonction utilise une mémoire pré-allouée basée sur les dimensions du flux vidéo initial, éliminant les temps d'allocation pendant leur exécution.

Par la suite, nous avons implanté des systèmes pour détecter et décrire les points d'intérêt. Nous avons opté pour le détecteur YAPE plutôt que FAST. En comparaison, le détecteur FAST et ORB d'OpenCV extraient un ensemble de points en environ 0,010 seconde par image. Notre implémentation, combinant YAPE et ORB, parvient à une extraction en 0,002 seconde en moyenne.

Nous avons utilisé l'algorithme Lucas-Kanade pour le suivi des points et avons mis en œuvre diverses optimisations pour améliorer les performances. Concernant les méthodes d'appariement, nous avons choisi une recherche par force brute en utilisant une distance de Hamming. Les algorithmes RANSAC et PROSAC ont été implémentés pour l'estimation d'homographie.

Suite au développement complet de notre bibliothèque, nous avons établi deux pipelines de suivi d'images pour une évaluation comparative avec OpenCV.

Le premier pipeline repose uniquement sur les algorithmes de détection. Pour chaque trame, l'analyse englobe l'intégralité de l'image, avec extraction et appariement des points d'intérêt.

Le second pipeline localise initialement le marqueur dans l'image, identifie un ensemble de points à suivre pour actualiser la position du marqueur, et se poursuit jusqu'à ce que le nombre de points soit insuffisant.

Ces pipelines ont validé les performances de notre bibliothèque avec des gains de l'ordre de 20 à

30 images par seconde suivant les appareils. En outre, la dimension du fichier wasm a été réduite de 5 Mo à 256 Ko, faisant probablement de notre solution l'une des applications Web de réalité augmentée les plus légères actuellement disponibles sur le marché.

5.5.3 Pipeline proposé

Ensuite, nous avons cherché à élaborer un pipeline algorithmique optimisé, ayant pour objectif de combiner les bénéfices des techniques de détection et de suivi d'images. De surcroît, une attention particulière est accordée à la limitation des opérations aux zones spécifiques de l'image.

La Figure 5.5 illustre notre proposition de pipeline. Cette conception vise principalement à localiser le marqueur de référence au sein d'un flux vidéo. Après avoir détecté ce dernier, la trame correspondante, ainsi que les données associées (par exemple, les descripteurs et l'homographie), sont stockées en tant qu'ancre.

Le processus se poursuit par le suivi de points spécifiques. Pour chaque succession de trames, nous déterminons la position d'un groupe prédéfini de points à suivre, s'appuyant sur l'homographie précédemment établie. Subséquemment, nous calculons leur déplacement en utilisant la méthode de Lucas-Kanade, et déduisons la transformation relative entre la trame antérieure et la trame courante.

Lorsque la transformation n'est pas clairement discernée, ou qu'un nombre insuffisant de points est correctement identifié, une tentative est faite pour faire correspondre la trame ancrée à la zone présumée du marqueur. Si cette tentative est infructueuse, une démarche similaire est entreprise, mais en utilisant les descripteurs de l'image de référence.

Si l'estimation basée sur Lucas-Kanade s'avère être fiable, avec un nombre de points excédant un certain seuil, il est alors nécessaire de mettre à jour l'homographie et d'évaluer la matrice extrinsèque. De plus, une fonctionnalité a été mise en place pour quantifier le niveau de mouvement observé ; si ce dernier dépasse un seuil prédéfini, une correction relative à l'ancre est appliquée. Suite à une validation réussie, les données relatives à l'ancre sont actualisées avec les informations de la trame actuelle.

Dans un premier temps, nous avons dû déterminer les meilleures combinaisons paramétriques, possibles afin

Afin d'évaluer la performance de notre proposition de pipeline, nous l'avons confrontée à divers pipelines dit naïfs. Ces derniers opèrent sur la base de divers mécanismes : purement par détection, par détection locale et en combinant détection et suivi de points d'intérêt.

Méthode	Erreur Pixels	Temps micro sec	Ratio image suivie
Détection	6.59	4.59	0.786
Détection—fenêtrée	6.10	4.17	0.84
Détection—Suivi	7.09	2.89	0.85
Notre pipeline	5.74	3.03	0.87

TABLE 5.1 – Résultats des vidéos avec des mouvements de translation

Le tableau 5.1 dévoile une convergence notable concernant les erreurs de mouvements de translation pour les quatre pipelines proposés. Pour le pipeline basé uniquement sur le suivi, une légère augmentation de l'erreur est observée, attribuée à une dérive des points suivis. De même, les pourcentages représentant la capacité des systèmes à fournir une position montrent peu de disparités, à l'ex-

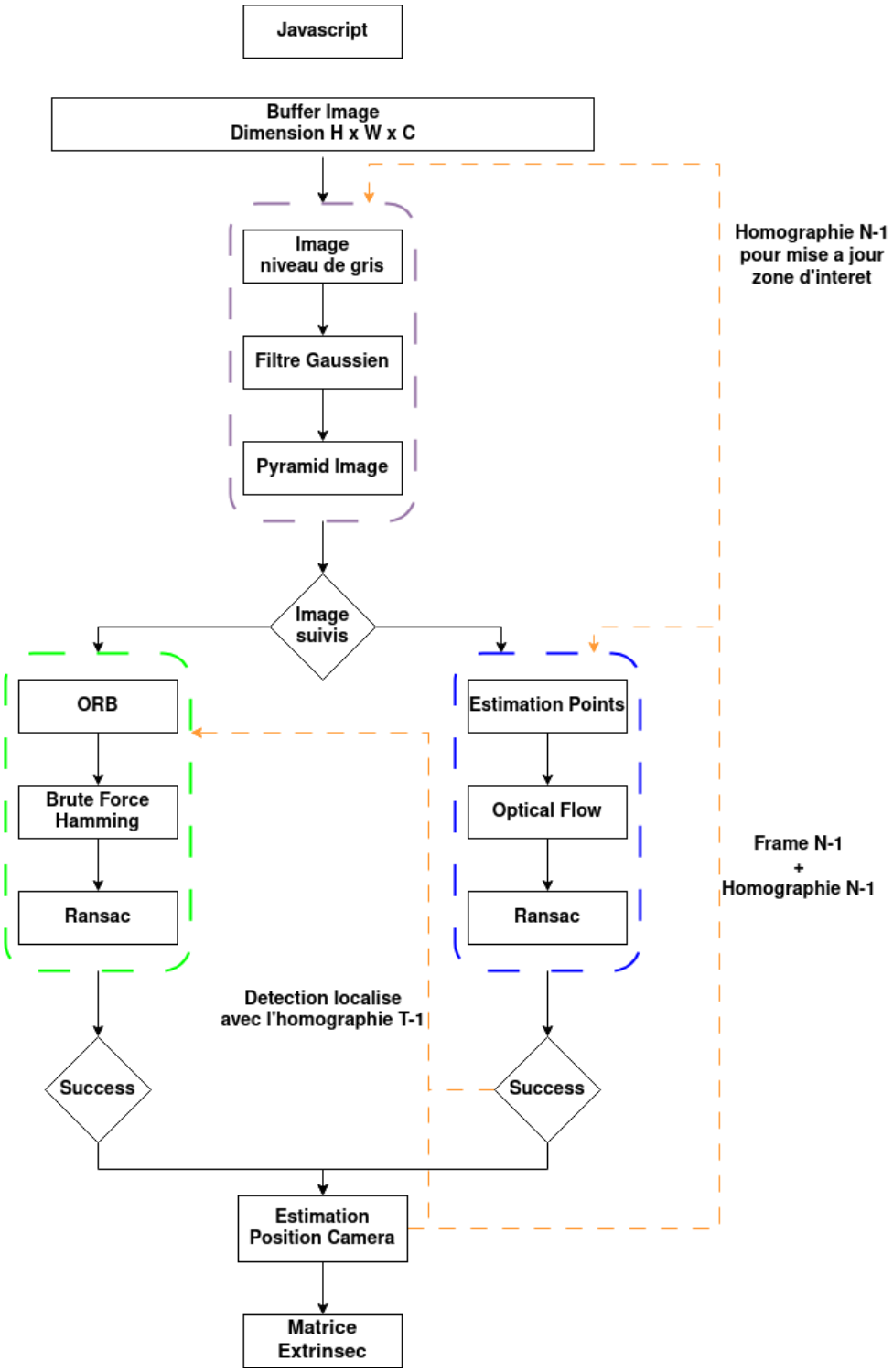


FIGURE 5.5 – Schéma du pipeline de suivi d'image

ception notable du pipeline basé sur une détection sans fenêtrage. Concernant les temps d'exécution, les méthodes employant Lucas-Kanade se distinguent par leurs performances supérieures.

Methode	Erreur Pixels	Temps micro sec	Ratio image suivie
Détection	10.22	4.20	0.67
Détection—fenêtrée	9.89	3.99	0.70
Détection—Suivi	14.54	2.61	0.99
Notre pipeline	11.67	2.79	0.99

TABLE 5.2 – Résultats des vidéos avec des mouvements perspectives

Lorsque confrontées à des mouvements comprenant des rotations, les limites des méthodes axées sur la détection deviennent manifestes. En effet, même avec la recherche fenêtrée, seules 70% des images sont correctement suivies. Les techniques utilisant Lucas-Kanade, en revanche, démontrent leur capacité à localiser une position à chaque trame, bien qu'au détriment de leur précision. Pour le pipeline dépourvu de correction, l'erreur observée atteint une moyenne alarmante de 14,54 pixels. L'absence de correction induit une dérive conséquente des points, compromettant grandement l'estimation de la position caméra.

Méthode	Erreur Pixels	Temps micro sec	Ratio image suivie
Détection	9.51	4.03	0.60
Détection—fenêtrée	7.73	4.07	0.69
Détection—Suivi	17.77	2.55	0.97
Notre pipeline	10.15	2.99	0.96

TABLE 5.3 – Résultats des vidéos avec des mouvements complexes

Dans des scénarios plus complexes, cette tendance est amplifiée. Le système de suivi s'appuyant sur Lucas-Kanade exhibe une erreur notablement supérieure aux autres méthodes. Nos analyses révèlent que notre pipeline maintient une précision comparable au système de détection, tout en offrant un taux de suivi des trames équivalent au système sans correction. De plus, il semble que le temps d'exécution, qui est un critère primordial, demeure constant.

La sélection d'un système optimal nécessite une calibration précise en ajustant divers paramètres pour accroître l'efficacité. Si bien que certains réglages peuvent aisément améliorer la précision, chaque modification risque d'impacter les temps de traitement. Afin d'assurer une expérience fluide, un mécanisme d'ajustement automatique des paramètres, fonctionnant en relation avec les performances du dispositif mobile, a été élaboré.

Les performances, en particulier le temps de calcul de notre pipeline, ont été examinées sur divers dispositifs et navigateurs. Les résultats sont prometteurs. Au moment de sa mise sur le marché, il se distinguait comme l'un des moteurs les plus rapides et précis disponibles. Toutefois, une baisse de performance est notable sur des appareils à faible puissance. Malgré cela, un taux de 30 images par seconde est maintenu sur la plupart des appareils, garantissant une expérience utilisateur satisfaisante.

Bien que la première version mise en œuvre soit fonctionnelle, des améliorations sont envisageables, notamment sur le plan de l'implémentation. Il est pertinent de noter que récemment, le compilateur

Appareil	Ips Détection	Ips Suivis	Navigateur
Ryzen 5 5600	90 fps	200 fps	Firefox
i7 Macbook pro 2018	80 fps	160 fps	Safari
Iphone 13	70 fps	140 fps	Safari
Iphone 8	23 fps	70 fps	Safari

TABLE 5.4 – Performances navigateurs Web en nombres d'images par seconde (Ips) pour différents appareils

Emscripten a intégré le support des instructions SIMD, une évolution par ailleurs reprise par plusieurs navigateurs contemporains.

Les instructions SIMD, acronyme de "Single Instruction, Multiple Data" en C++, constituent un jeu d'instructions spécifiquement conçu pour effectuer des opérations simultanées sur plusieurs données. En d'autres termes, une unique instruction SIMD a la capacité de traiter de multiples entrées simultanément, contrairement à une approche traditionnelle qui traiterait chaque donnée de manière séquentielle. Ces instructions se révèlent particulièrement avantageuses pour des calculs intensifs, notamment les calculs vectoriels et matriciels.

La mise en œuvre récente de certaines de ces instructions vise à optimiser diverses fonctions, comme l'application de filtres ou le calcul de la distance de Hamming. Cette intégration permet d'exploiter efficacement le parallélisme intrinsèque aux architectures de processeurs modernes, aboutissant à des gains variables.

5.6 Bilan

Ce chapitre a abordé le domaine des applications de réalité augmentée fondées sur des marqueurs, avec une attention particulière portée aux images planes. La première version du moteur de suivi d'images que nous avons conçues en C++ et ultérieurement compilé en WebAssembly a été exposée en profondeur. Grâce à ces architectures, des expériences de réalité augmentée ont pu être mises en œuvre pour des applications Web. Le recours au format WebAssembly assure une compatibilité étendue avec une diversité d'appareils mobiles. De plus, l'absence de dépendance à une bibliothèque externe limite le poids du fichier binaire à 240 Ko lors de chaque initialisation de l'application Web. Il est à noter que ce moteur est actuellement exploité et commercialisé par la société ARGO.

Par ailleurs, nous avons entrepris le développement d'une version du moteur capable d'identifier des images sur des surfaces cylindriques, à l'instar des bouteilles. La principale distinction de cette variante repose sur la méthodologie d'estimation de la position de la caméra, laquelle s'appuie sur un calcul de Perspective-n-Point (PnP). Néanmoins, cette version se révèle plus chronophage du fait des calculs additionnels requis pour évaluer cette position à partir de données tridimensionnelles. De ce fait, cette itération n'a pas encore été mise sur le marché à ce jour.

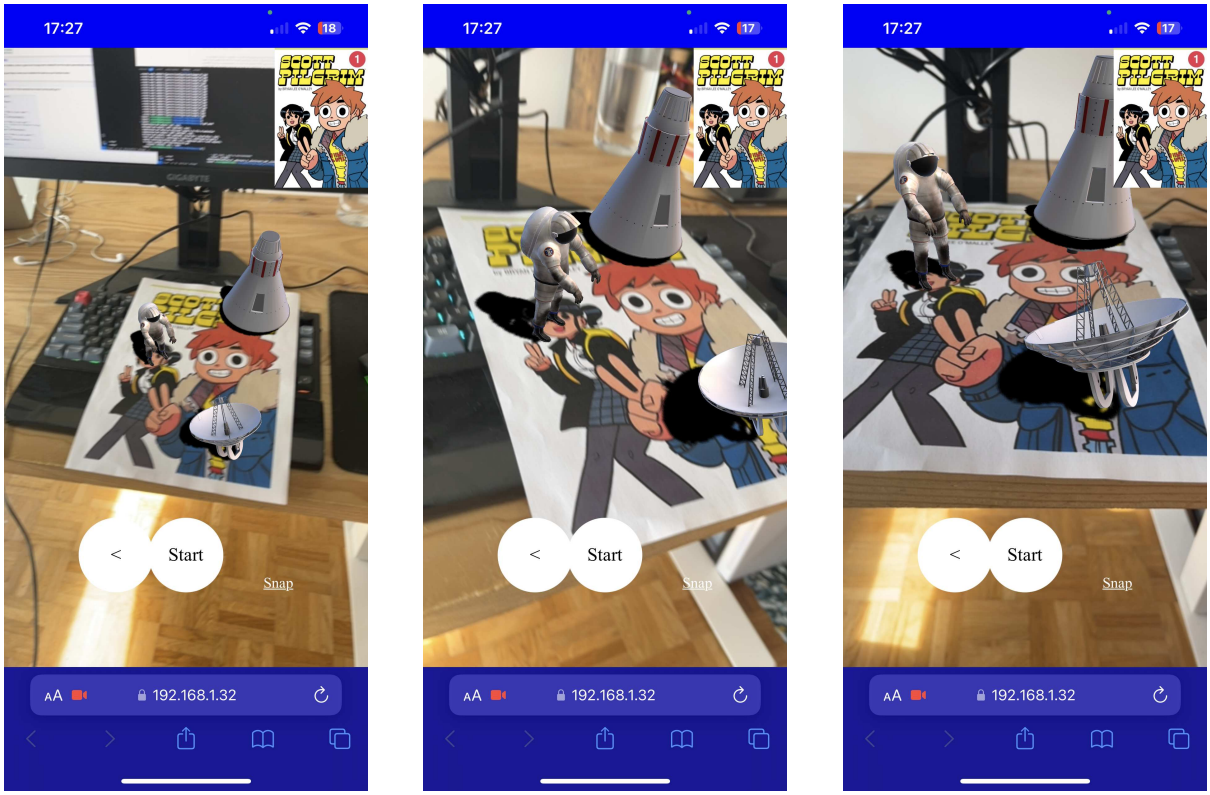


FIGURE 5.6 – Rendu final dans l'application sur un iPhone 13

Chapitre 6

Conclusion

6.1 Bilan

Dans ce manuscrit, nous avons étudié les étapes clés du fonctionnement d'une application de réalité augmentée basée sur des marqueurs. Chaque étape soulève ses propres défis. La première concerne la détection et l'identification des marqueurs visuels spécifiques via la caméra d'un appareil. Une fois identifiés, ces marqueurs deviennent des repères pour superposer des éléments virtuels, tels que des images, vidéos ou modèles 3D, à la vue réelle de l'environnement.

La deuxième étape traite de l'estimation en temps réel de la position et de l'orientation du marqueur, assurant l'alignement des éléments virtuels avec la réalité pendant les mouvements de l'appareil. Les utilisateurs peuvent ensuite interagir avec ces éléments par des gestes et des commandes tactiles. Cette technologie offre une expérience immersive, fusionnant de façon harmonieuse réalité et virtuel.

Deux problématiques industrielles ont été identifiées : la reconnaissance des marqueurs et le suivi en temps réel sur des appareils moyennes gammes dans des environnements Web. Nous avons d'abord abordé différentes méthodes liées à la construction de descripteurs globaux et locaux, en explorant plusieurs pipelines d'évaluation pour comprendre les problématiques relatifs à l'identification et aux confusions de marqueur.

Ensuite, nous avons mis en avant nos innovations, proposant une méthode pour atténuer ce problème, notamment via un système de détection et un nouveau pipeline de reconnaissance d'images. Cette approche divise les images en sous-sections, visant à minimiser les données redondantes et, par conséquent, les confusions. De plus, la recherche par fenêtrage et la méthodologie utilise pour valider le résultat permet une meilleure identification, avec des résultats pouvant obtenir une précision bien supérieure et pouvant s'approcher d'un système d'identification.

La collaboration avec une entreprise privée nous a imposé plusieurs contraintes. Pour les systèmes de reconnaissance, l'enjeu était d'atteindre une réactivité optimale (environ une seconde) tout en maximisant les processus côté utilisateur, qu'il s'agisse d'applications natives ou web. Quant au suivi d'images, l'exigence était de développer une bibliothèque propriétaire compatible avec le plus grand nombre d'appareils, ce qui excluait les approches par apprentissage.

Face à ces défis industriels notables, nous avons conçu des systèmes désormais commercialisés (tels que notre bibliothèque en C++) et d'autres en phase d'implémentation finale (notamment notre moteur de reconnaissance d'images). Ces progrès représentent un atout commercial significatif pour la société ARGO. Sur le plan académique, nos recherches ont abouti à la rédaction de deux articles : le premier introduisant une nouvelle base de données dédiée à notre thématique, et le second traitant de notre système optimisé de détection de documents pour une utilisation web. Nous avons par ailleurs suggéré une méthodologie innovante pour la recherche d'images ; toutefois, celle-ci n'en est qu'à ses prémices et requiert des investigations et des optimisations supplémentaires qui pourraient par la suite déboucher à la proposition d'un article.

6.2 Perspectives

Suite à l'implémentation de notre nouveau pipeline, de nouvelles perspectives de recherche émergent. La première concerne l'approche adoptée pour le traitement des images de référence. Actuellement, notre méthode procède à une segmentation rigoureuse des images. Une orientation de recherche pertinente consisterait à automatiser ce processus en le rendant plus adaptatif. Par exemple, des systèmes

de segmentation pourraient être employés pour extraire de manière sélective des éléments textuels et graphiques et donc définir automatiquement les sous-images. Une autre piste prometteuse serait l'isolation automatique des zones saillantes au sein de collections d'images présentant des caractéristiques communes.

Un autre domaine d'intérêt est la génération de vecteurs descriptifs singuliers. Notre architecture actuelle est conçue pour s'adapter à toute méthode produisant un vecteur distinct pour chaque image. Dans cette optique, il conviendrait d'étudier différentes stratégies, telles que l'adoption de descripteurs plus compacts ou l'évaluation de Visual Transformers (ViT [El-+21]) qui ont montré des résultats intéressants, mais nécessitant une grande quantité de données et un coût d'entrainement important.

Concernant le suivi d'images en temps réel, une profusion de travaux est recensée dans la littérature académique faisant appel à des méthodologies d'apprentissage profond. Néanmoins, ces techniques s'avèrent souvent inadaptées pour des déploiements web sur des dispositifs de gamme intermédiaire. Cependant, il convient de noter, que certains équipements récents sont équipés de dispositifs dotés d'accélérateurs matériels dédiés à l'apprentissage profond, tels que les TPU. Ainsi, il serait judicieux d'anticiper cette évolution et de considérer la conception d'une architecture allégée et optimisée, similaire en essence à notre système actuel de détection de document.

Bibliographie

[Aba+16] Martín ABADI et al. “Tensorflow : A system for large-scale machine learning”. In : *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016, p. 265-283.

[AOV12] Alexandre ALAHI, Raphael ORTIZ et Pierre VANDERGHEYNST. “Freak : Fast retina keypoint”. In : *2012 IEEE conference on computer vision and pattern recognition*. Ieee. 2012, p. 510-517.

[Ara+18] Teresa ARAÚJO et al. “UOLO-automatic object detection and segmentation in biomedical images”. In : *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support : 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*. Springer. 2018, p. 165-173.

[AZ13] Relja ARANDJELOVIC et Andrew ZISSERMAN. “All about VLAD”. In : *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2013, p. 1578-1585.

[Bay+08] Herbert BAY et al. “Speeded-up robust features (SURF)”. In : *Computer vision and image understanding* 110.3 (2008), p. 346-359.

[Bea78] Paul R BEAUDET. “Rotational invariant image operators”. In : *Proc. 4th International Joint Conference on Pattern Recognition (ICPR)*. 1978, p. 579-583.

[Bur+15] Jean-Christophe BURIE et al. “ICDAR2015 competition on smartphone document capture and OCR (SmartDoc)”. In : *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2015, p. 1161-1165.

[Cal+10] Michael CALONDER et al. “Brief : Binary robust independent elementary features”. In : *Computer Vision–ECCV 2010 : 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer. 2010, p. 778-792.

[CAS20] Bingyi CAO, Andre ARAUJO et Jack SIM. “Unifying deep local and global features for image search”. In : *Computer Vision–ECCV 2020 : 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer International Publishing. 2020, p. 726-743.

[Cha+10] Savvas A CHATZICHRISTOFIS et al. “Accurate image retrieval based on compact composite descriptors and relevance feedback information”. In : *International Journal of Pattern Recognition and Artificial Intelligence* 24.02 (2010), p. 207-244.

[Cha+11] Vijay R CHANDRASEKHAR et al. “The stanford mobile visual search data set”. In : *Proceedings of the second annual ACM conference on Multimedia systems*. 2011, p. 117-122.

[Cho17] François CHOLLET. “Xception : Deep learning with depthwise separable convolutions”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 1251-1258.

[CM05] Ondrej CHUM et Jiri MATAS. “Matching with PROSAC-progressive sample consensus”. In : *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. T. 1. IEEE. 2005, p. 220-226.

[Com18] Blender Online COMMUNITY. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL : http://www.blender.org.

[Csu+04] Gabriella CSURKA et al. “Visual categorization with bags of keypoints”. In : *Workshop on statistical learning in computer vision, ECCV*. T. 1. 1-22. Prague. 2004, p. 1-2.

[Dan+14] Quoc Bao DANG et al. “A multi-layer approach for camera-based complex map image retrieval and spotting system”. In : *2014 4th International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE. 2014, p. 1-6.

[Dan+15a] Quoc Bao DANG et al. “Camera-based document image retrieval system using local features-comparing SRIF with LLAH, SIFT, SURF and ORB”. In : *2015 13th International conference on document analysis and recognition (ICDAR)*. IEEE. 2015, p. 1211-1215.

[Dan+15b] Quoc Bao DANG et al. “Srif : Scale and rotation invariant features for camera-based document image retrieval”. In : *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2015, p. 601-605.

[Dan+16] Quoc Bao DANG et al. “Camera-based document image spotting system for complex linguistic maps”. In : *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2016, p. 003246-003251.

[Dan+18] Quoc Bao DANG et al. “New spatial-organization-based scale and rotation invariant features for heterogeneous-content camera-based document image retrieval”. In : *Pattern Recognition Letters* 112 (2018), p. 153-160.

[Dan+19] Quoc Bao DANG et al. “A comparison of local features for camera-based document image retrieval and spotting”. In : *International Journal on Document Analysis and Recognition (IJDAR)* 22 (2019), p. 247-263.

[Del+13] Jonathan DELHUMEAU et al. “Revisiting the VLAD image representation”. In : *Proceedings of the 21st ACM international conference on Multimedia*. 2013, p. 653-656.

[Den+19] Jiankang DENG et al. “Arcface : Additive angular margin loss for deep face recognition”. In : *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, p. 4690-4699.

[DHS73] Richard O DUDA, Peter E HART et David G STORK. *Pattern classification and scene analysis*. T. 3. Wiley New York, 1973.

[DMR18] Daniel DETONE, Tomasz MALISIEWICZ et Andrew RABINOVICH. “Superpoint : Self-supervised interest point detection and description”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, p. 224-236.

[El-+21] Alaaeldin EL-NOUBY et al. “Training vision transformers for image retrieval”. In : *arXiv preprint arXiv :2102.05644* (2021).

[ERL14] Christian EGGERT, Stefan ROMBERG et Rainer LIENHART. “Improving VLAD : hierarchical coding and a refined local coordinate system”. In : *2014 IEEE international conference on image processing (ICIP)*. IEEE. 2014, p. 3018-3022.

[FB81] Martin A FISCHLER et Robert C BOLLES. “Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography”. In : *Communications of the ACM* 24.6 (1981), p. 381-395.

[FBF77] Jerome H FRIEDMAN, Jon Louis BENTLEY et Raphael Ari FINKEL. “An algorithm for finding best matches in logarithmic expected time”. In : *ACM Transactions on Mathematical Software (TOMS)* 3.3 (1977), p. 209-226.

[Fli+95] Myron FLICKNER et al. “Query by image and video content : The QBIC system”. In : *computer* 28.9 (1995), p. 23-32.

[Gir15] Ross GIRSHICK. “Fast r-cnn”. In : *Proceedings of the IEEE international conference on computer vision*. 2015, p. 1440-1448.

[Gor+16] Albert GORDO et al. “Deep image retrieval : Learning global representations for image search”. In : *Computer Vision–ECCV 2016 : 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer. 2016, p. 241-257.

[He+16] Kaiming HE et al. “Deep residual learning for image recognition”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 770-778.

[He+17] Kaiming HE et al. “Mask r-cnn”. In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 2961-2969.

[HS+88] Chris HARRIS, Mike STEPHENS et al. “A combined corner and edge detector”. In : *Alvey vision conference*. T. 15. 50. Citeseer. 1988, p. 10-5244.

[HSS18] Jie HU, Li SHEN et Gang SUN. “Squeeze-and-excitation networks”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, p. 7132-7141.

[Hul+07] Jonathan J HULL et al. “Based augmented reality”. In : *17th International Conference on Artificial Reality and Telexistence (ICAT 2007)*. IEEE. 2007, p. 205-209.

[HZ93] Geoffrey E HINTON et Richard ZEMEL. “Autoencoders, minimum description length and Helmholtz free energy”. In : *Advances in neural information processing systems* 6 (1993).

[IM98] Piotr INDYK et Rajeev MOTWANI. “Approximate nearest neighbors : towards removing the curse of dimensionality”. In : *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 1998, p. 604-613.

[INK07] Masakazu IWAMURA, Tomohiro NAKAI et Koichi KISE. “Improvement of Retrieval Speed and Required Amount of Memory for Geometric Hashing by Combining Local Invariants.” In : *BMVC*. 2007, p. 1-10.

[JDJ19] Jeff JOHNSON, Matthijs DOUZE et Hervé JÉGOU. “Billion-scale similarity search with GPUs”. In : *IEEE Transactions on Big Data* 7.3 (2019), p. 535-547.

[Jég+10] Hervé JÉGOU et al. “Aggregating local descriptors into a compact image representation”. In : *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, p. 3304-3311.

[JS17] Khurram JAVED et Faisal SHAFAIT. “Real-time document localization in natural images by recursive application of a cnn”. In : *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. T. 1. IEEE. 2017, p. 105-110.

[JSZ+15] Max JADERBERG, Karen SIMONYAN, Andrew ZISSERMAN et al. “Spatial transformer networks”. In : *Advances in neural information processing systems* 28 (2015).

[KK14] Philipp KRÄHENBÜHL et Vladlen KOLTUN. “Geodesic object proposals”. In : *European conference on computer vision*. Springer. 2014, p. 725-739.

[KSH17] Alex KRIZHEVSKY, Ilya SUTSKEVER et Geoffrey E HINTON. “Imagenet classification with deep convolutional neural networks”. In : *Communications of the ACM* 60.6 (2017), p. 84-90.

[LB16] Luciano RS LEAL et Byron LD BEZERRA. “Smartphone camera document detection via Geodesic Object Proposals”. In : *2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. IEEE. 2016, p. 1-6.

[LCS11] Stefan LEUTENEGGER, Margarita CHLI et Roland Y SIEGWART. “BRISK : Binary robust invariant scalable keypoints”. In : *2011 International conference on computer vision*. Ieee. 2011, p. 2548-2555.

[LD07] Xu LIU et David DOERMANN. “Mobile retriever-finding document with a snapshot”. In : *Int. Workshop on Camera-Based Document Analysis and Recognition*. 2007, p. 29-34.

[LeC+98] Yann LECUN et al. “Gradient-based learning applied to document recognition”. In : *Proceedings of the IEEE* 86.11 (1998), p. 2278-2324.

[Lee+22] Seongwon LEE et al. “Correlation verification for image retrieval”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, p. 5374-5384.

[Leu+15] Stefan LEUTENEGGER et al. “Keyframe-based visual–inertial odometry using nonlinear optimization”. In : *The International Journal of Robotics Research* 34.3 (2015), p. 314-334.

[Lin98] Tony LINDEBERG. “Feature detection with automatic scale selection”. In : *International journal of computer vision* 30.2 (1998), p. 79-116.

[Llo82] Stuart LLOYD. “Least squares quantization in PCM”. In : *IEEE transactions on information theory* 28.2 (1982), p. 129-137.

[LM13] Mingyang LI et Anastasios I MOURIKIS. “High-precision, consistent EKF-based visual-inertial odometry”. In : *The International Journal of Robotics Research* 32.6 (2013), p. 690-711.

[LNK18] A LUKOYANOV, D NIKOLAEV et Ivan KONOVALENKO. “Modification of YAPE keypoint detection algorithm for wide local contrast range images”. In : *Tenth International Conference on Machine Vision (ICMV 2017)*. T. 10696. SPIE. 2018, p. 305-312.

[Low04] David G LOWE. “Distinctive image features from scale-invariant keypoints”. In : *International journal of computer vision* 60 (2004), p. 91-110.

[Low99] David G LOWE. “Object recognition from local scale-invariant features”. In : *Proceedings of the seventh IEEE international conference on computer vision*. T. 2. Ieee. 1999, p. 1150-1157.

[LPZ22a] Thibault LELONG, Marius PREDA et Titus ZAHARIA. “A new database for image retrieval of camera filmed printed documents”. In : *Proceedings of the 27th International Conference on 3D Web Technology*. 2022, p. 1-4.

[LPZ22b] Thibault LELONG, Marius PREDA et Titus ZAHARIA. “Document Segmentation for WebAR application”. In : *Proceedings of the 27th International Conference on 3D Web Technology*. 2022, p. 1-4.

[LSP23] Philipp LINDENBERGER, Paul-Edouard SARLIN et Marc POLLEFEYS. “LightGlue : Local Feature Matching at Light Speed”. In : *arXiv preprint arXiv :2306.13643* (2023).

[LT08] Shijian LU et Chew Lim TAN. “Retrieval of machine-printed latin documents through word shape coding”. In : *Pattern Recognition* 41.5 (2008), p. 1799-1809.

[Ma+18] Ke MA et al. “Docunet : Document image unwarping via a stacked u-net”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, p. 4700-4709.

[Mac67] J. MACQUEEN. “Some methods for classification and analysis of multivariate observations”. In : 1967.

[Mai+10] Elmar MAIR et al. “Adaptive and generic corner detection based on the accelerated segment test”. In : *Computer Vision–ECCV 2010 : 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*. Springer. 2010, p. 183-196.

[Min+20] Weiqing MIN et al. “A two-stage triplet network training framework for image retrieval”. In : *IEEE Transactions on Multimedia* 22.12 (2020), p. 3128-3138.

[Møl18] Anders MØLLER. “Technical perspective : WebAssembly : A quiet revolution of the Web”. In : *Communications of the ACM* 61.12 (2018), p. 106-106.

[Mor77] Hans P. MORAVEC. “Towards Automatic Visual Obstacle Avoidance”. In : *International Joint Conference on Artificial Intelligence*. 1977.

[MP43] Warren S MCCULLOCH et Walter PITTS. “A logical calculus of the ideas immanent in nervous activity”. In : *The bulletin of mathematical biophysics* 5 (1943), p. 115-133.

[MS01] Krystian MIKOLAJCZYK et Cordelia SCHMID. “Indexing based on scale invariant interest points”. In : *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. T. 1. IEEE. 2001, p. 525-531.

[MS04] Krystian MIKOLAJCZYK et Cordelia SCHMID. “Scale & affine invariant interest point detectors”. In : *International journal of computer vision* 60 (2004), p. 63-86.

[MS05] Krystian MIKOLAJCZYK et Cordelia SCHMID. “A performance evaluation of local descriptors”. In : *IEEE transactions on pattern analysis and machine intelligence* 27.10 (2005), p. 1615-1630.

[Nev+20] Ricardo Batista das NEVES JUNIOR et al. “HU-PageScan : a fully convolutional neural network for document page crop”. In : *IET Image Processing* 14.15 (2020), p. 3890-3898.

[NFG19] Minh Ôn Vũ NGOC, Jonathan FABRIZIO et Thierry GÉRAUD. “Document detection in videos captured by smartphones using a saliency-based method”. In : *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*. T. 4. IEEE. 2019, p. 19-24.

[NKI05] Tomohiro NAKAI, Koichi KISE et Masakazu IWAMURA. “Hashing with local combinations of feature points and its application to camera-based document image retrieval”. In : *Proc. CBDAR05* (2005), p. 87-94.

[NKI06] Tomohiro NAKAI, Koichi KISE et Masakazu IWAMURA. “Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval”. In : *Document Analysis Systems VII : 7th International Workshop, DAS 2006, Nelson, New Zealand, February 13-15, 2006. Proceedings 7*. Springer. 2006, p. 541-552.

[NKI07] Tomohiro NAKAI, Koichi KISE et Masakazu IWAMURA. “Camera based document image retrieval with more time and memory efficient LLAH”. In : *Proc. CBDAR* (2007), p. 21-28.

[NKI09] Tomohiro NAKAI, Koichi KISE et Masakazu IWAMURA. “Real-time retrieval for images of documents in various languages using a web camera”. In : *2009 10th International Conference on Document Analysis and Recognition*. IEEE. 2009, p. 146-150.

[Noh+17] Hyeonwoo NOH et al. “Large-scale image retrieval with attentive deep local features”. In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 3456-3465.

[NV96] François NORMANT et Axel VAN DE WALLE. “The sausage of local convex hulls of a curve and the Douglas-Peucker algorithm”. In : *Cartographica : the International Journal for Geographic Information and Geovisualization* 33.4 (1996), p. 25-34.

[OD11] Stephen O’HARA et Bruce A DRAPER. “Introduction to the bag of features paradigm for image classification and retrieval”. In : *arXiv preprint arXiv :1101.3354* (2011).

[OT01] Aude OLIVA et Antonio TORRALBA. “Modeling the shape of the scene : A holistic representation of the spatial envelope”. In : *International journal of computer vision* 42 (2001), p. 145-175.

[Pas+12] Stavros PASCHALAKIS et al. “The MPEG-7 video signature tools for content identification”. In : *IEEE transactions on circuits and systems for video technology* 22.7 (2012), p. 1050-1063.

[Pas+19] Adam PASZKE et al. “PyTorch : An Imperative Style, High-Performance Deep Learning Library”. In : *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, p. 8024-8035. URL : http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[PSM10] Florent PERRONNIN, Jorge SÁNCHEZ et Thomas MENSINK. “Improving the fisher kernel for large-scale image classification”. In : *Computer Vision–ECCV 2010 : 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer. 2010, p. 143-156.

[Qia+19] Xiuquan QIAO et al. “Web AR : A promising future for mobile augmented reality—State of the art, challenges, and insights”. In : *Proceedings of the IEEE* 107.4 (2019), p. 651-666.

[RD06] Edward ROSTEN et Tom DRUMMOND. “Machine learning for high-speed corner detection”. In : *Computer Vision–ECCV 2006 : 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer. 2006, p. 430-443.

[Red+16] Joseph REDMON et al. “You only look once : Unified, real-time object detection”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 779-788.

[RFB15] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX. “U-net : Convolutional networks for biomedical image segmentation”. In : *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, p. 234-241.

[RK87] LKPJ RDUSSEEUN et P KAUFMAN. “Clustering by means of medoids”. In : *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*. T. 31. 1987.

[Ros58] Frank ROSENBLATT. “The perceptron : a probabilistic model for information storage and organization in the brain.” In : *Psychological review* 65.6 (1958), p. 386.

[RTC16] Filip RADENOVIĆ, Giorgos TOLIAS et Ondřej CHUM. “CNN image retrieval learns from BoW : Unsupervised fine-tuning with hard examples”. In : *Computer Vision–ECCV 2016 : 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, p. 3-20.

[Rub+11] Ethan RUBLEE et al. “ORB : An efficient alternative to SIFT or SURF”. In : *2011 International conference on computer vision*. Ieee. 2011, p. 2564-2571.

[Rus+15] Olga RUSSAKOVSKY et al. “Imagenet large scale visual recognition challenge”. In : *International journal of computer vision* 115 (2015), p. 211-252.

[Sar+20] Paul-Edouard SARLIN et al. “Superglue : Learning feature matching with graph neural networks”. In : *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, p. 4938-4947.

[SB91] Michael J SWAIN et Dana H BALLARD. “Color indexing”. In : *International journal of computer vision* 7.1 (1991), p. 11-32.

[SB97] Stephen M SMITH et J Michael BRADY. “SUSAN—a new approach to low level image processing”. In : *International journal of computer vision* 23.1 (1997), p. 45-78.

[Shr+17] Ashish SHRIVASTAVA et al. “Learning from simulated and unsupervised images through adversarial training”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 2107-2116.

[Sko+15] Natalya SKORYUKINA et al. “Real time rectangular document detection on mobile devices”. In : *Seventh International Conference on Machine Vision (ICMV 2014)*. T. 9445. International Society for Optics et Photonics. 2015, 94452A.

[Suz+85] Satoshi SUZUKI et al. “Topological structural analysis of digitized binary images by border following”. In : *Computer vision, graphics, and image processing* 30.1 (1985), p. 32-46.

[SZ14] Karen SIMONYAN et Andrew ZISSERMAN. “Very deep convolutional networks for large-scale image recognition”. In : *arXiv preprint arXiv :1409.1556* (2014).

[Sze+15] Christian SZEGEDY et al. “Going deeper with convolutions”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, p. 1-9.

[Sze+16] Christian SZEGEDY et al. “Rethinking the inception architecture for computer vision”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 2818-2826.

[Sze+17] Christian SZEGEDY et al. “Inception-v4, inception-resnet and the impact of residual connections on learning”. In : *Proceedings of the AAAI conference on artificial intelligence*. T. 31. 1. 2017.

[Tah+17] Sajjad TAHERI et al. “OpenCV. js : Computer vision processing for the Web”. In : *Univ. California, Irvine, Irvine, CA, USA, Tech. Rep* (2017).

[Tah+18] Sajjad TAHERI et al. “Opencv. js : Computer vision processing for the open web platform”. In : *Proceedings of the 9th ACM Multimedia Systems Conference*. 2018, p. 478-483.

[TJC20] Giorgos TOLIAS, Tomas JENICEK et Ondřej CHUM. “Learning and aggregating deep local descriptors for instance-level recognition”. In : *Computer Vision–ECCV 2020 : 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, p. 460-477.

[TKI11] Kazutaka TAKEDA, Koichi KISE et Masakazu IWAMURA. “Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved llah”. In : *2011 International Conference on Document Analysis and Recognition*. IEEE. 2011, p. 1054-1058.

[TKI12] Kazutaka TAKEDA, Koichi KISE et Masakazu IWAMURA. “Real-time document image retrieval on a smartphone”. In : *2012 10th IAPR International Workshop on Document Analysis Systems*. IEEE. 2012, p. 225-229.

[Tro+20] Daniil V TROPIN et al. “Approach for document detection by contours and contrasts”. In : *arXiv preprint arXiv :2008.02615* (2020).

[Var+17] Gul VAROL et al. “Learning from synthetic humans”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 109-117.

[Von+12] Rafael Grompone VON GIOI et al. “LSD : A line segment detector”. In : *Image Processing On Line* 2 (2012), p. 35-55.

[Wan+14] Ji WAN et al. “Deep learning for content-based image retrieval : A comprehensive study”. In : *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, p. 157-166.

[Wan+17] Fei WANG et al. “Residual attention network for image classification”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 3156-3164.

[Won+16] Sebastien C WONG et al. “Understanding data augmentation for classification : when to warp ?” In : *2016 international conference on digital image computing : techniques and applications (DICTA)*. IEEE. 2016, p. 1-6.

[Woo+18] Sanghyun WOO et al. “Cbam : Convolutional block attention module”. In : *Proceedings of the European conference on computer vision (ECCV)*. 2018, p. 3-19.

[Wu+19] Huikai WU et al. “Fastfcn : Rethinking dilated convolution in the backbone for semantic segmentation”. In : *arXiv preprint arXiv :1903.11816* (2019).

[Wu+23] Han WU et al. “LDRNet : Enabling Real-time Document Localization on Mobile Devices”. In : *Machine Learning and Principles and Practice of Knowledge Discovery in Databases : International Workshops of ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part I*. Springer. 2023, p. 618-629.

[Xia+19] Xuanlu XIANG et al. “Multiple saliency and channel sensitivity network for aggregated convolutional feature”. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 33. 01. 2019, p. 9013-9020.

[Xie+21] Guo-Wang XIE et al. “Document dewarping with control points”. In : *Document Analysis and Recognition–ICDAR 2021 : 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16*. Springer. 2021, p. 466-480.

[Xu+16] Yan XU et al. “Improved relation classification by deep recurrent neural networks with data augmentation”. In : *arXiv preprint arXiv :1601.03651* (2016).

[Xue+22] Chuhui XUE et al. “Fourier document restoration for robust document dewarping and recognition”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, p. 4573-4582.

[Yan+21] Min YANG et al. “Dolg : Single-stage image retrieval with deep orthogonal fusion of local and global features”. In : *Proceedings of the IEEE/CVF International conference on Computer Vision*. 2021, p. 11772-11781.

[Yan87] L YANN. “Modeles connexionnistes de lapprentissage”. Thèse de doct. These de Doctorat, Universite Paris, 1987.

[Zag+10] Konstantinos ZAGORIS et al. “Automatic image annotation and retrieval using the joint composite descriptor”. In : *2010 14th Panhellenic Conference on Informatics*. IEEE. 2010, p. 143-147.

[Zha+17] Wenxiao ZHANG et al. “Cloudar : A cloud-based framework for mobile augmented reality”. In : *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. 2017, p. 194-200.

ECOLE DOCTORALE

Titre : Reconnaissance des documents avec de l'apprentissage profond pour la réalité augmentée

Mots clés : Reconnessaince d'image, apprentissage profond, suivi d'image, réalité augmentée

Résumé :
Ce projet doctoral se focalise sur les problématiques associées à l'identification d'images et de documents dans les applications de réalité augmentée utilisant des marqueurs, en particulier lors de l'utilisation d'appareils photo. La recherche s'inscrit dans un contexte technologique où l'interaction via la réalité augmentée est essentielle dans plusieurs domaines, y compris l'industrie, qui requièrent des méthodologies d'identification fiables.
Dans une première phase, le projet évalue diverses méthodologies d'identification et de traitement d'image au moyen d'une base de données spécialement conçue pour refléter les défis du contexte industriel. Cette recherche permet une analyse approfondie des méthodologies existantes, révélant ainsi leurs potentiels et leurs limites dans divers scénarios d'application.
Par la suite, le projet propose un système de détection de documents visant à améliorer les solutions existantes, optimisé pour des environnements tels que les navigateurs web. Ensuite, une méthodologie innovante pour la recherche d'images est introduite, s'appuyant sur une analyse de l'image en sous-parties afin d'accroître la précision de l'identification et d'éviter les confusions d'images. Cette approche permet une identification plus précise et adaptative, notamment en ce qui concerne les variations de la mise en page de l'image cible.
Enfin, dans le cadre de travaux en collaboration avec la société ARGO, un moteur de suivi d'image en temps réel a été développé, optimisé pour des appareils à basse puissance et pour les environnements web. Ceci assure le déploiement d'applications web en réalité augmentée et leur fonctionnement sur un large éventail de dispositifs, y compris ceux dotés de capacités de traitement limitées.
Il est à noter que les travaux issus de ce projet doctoral ont été appliqués et valorisés concrètement par la société Argo à des fins commerciales, confirmant ainsi la pertinence et la viabilité des méthodologies et solutions développées, et attestant de leur contribution significative au domaine technologique et industriel de la réalité augmentée.

Title : Document recognition with deep learning for augmented reality

Keywords : Image recognition, deep learning, image tracking, augmented reality

Abstract :
This doctoral project focuses on issues related to the identification of images and documents in augmented reality applications using markers, particularly when using cameras. The research is set in a technological context where interaction through augmented reality is essential in several domains, including industry, which require reliable identification methodologies.
In an initial phase, the project assesses various identification and image processing methodologies using a database specially designed to reflect the challenges of the industrial context. This research allows an in-depth analysis of existing methodologies, thus revealing their potentials and limitations in various application scenarios.
Subsequently, the project proposes a document detection system aimed at enhancing existing solutions, optimized for environments such as web browsers. Then, an innovative image research methodology is introduced, relying on an analysis of the image in sub-parts to increase the accuracy of identification and avoid image confusions. This approach allows for more precise and adaptive identification, particularly with respect to variations in the layout of the target image.
Finally, in the context of collaborative work with ARGO company, a real-time image tracking engine was developed, optimized for low-power devices and web environments. This ensures the deployment of augmented reality web applications and their operation on a wide range of devices, including those with limited processing capabilities.
It is noteworthy that the works resulting from this doctoral project have been concretely applied and valorized by the Argo company for commercial purposes, thereby confirming the relevance and viability of the developed methodologies and solutions, and attesting to their significant contribution to the technological and industrial field of augmented reality.

Institut Polytechnique de Paris
91120 Palaiseau, France