



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg



Projet de semestre 6

Filière télécommunications, orientation internet et communication

Medical Machine Learning

Rapport

Version 7.1

Rédigé par

Patrick Audriaz

Superviseurs

Andreas Fischer, Nicolas Schroeter

Mandant

Jérôme Clément (e-sculape)

Fribourg, 8 mai 2019

Copyright © 2019 Patrick Audriaz
Haute école d'ingénierie et d'architecture Fribourg
LaTeX-template-for-Word by sebnil on GitHub

DOCUMENT

1 Métadonnées

Auteur : Patrick Audriaz

Superviseurs : Andreas Fischer, Nicolas Schroeter

Mandant : Jérôme Clément (e-sculape)

Date d'édition : 8 mai 2019

Version : 7.1

2 Organisation et liens

Tous les documents concernant le projet sont déposés sur la forge à l'adresse :

<https://forge.hefr.ch/projects/medical-machine-learning>

Tous les fichiers du projet sont déposés sur le dépôt Git disponible à l'adresse suivante :

<https://gitlab.forge.hefr.ch/patrick.audriaz/ps6-audriaz>

Les modèles CNN de Deep Learning se trouvent à l'adresse suivante :

<https://www.kaggle.com/patrickaudriaz/kernels>

3 Table des versions

Version	Date	Remarque
1.0	6.3.2019	Création du document
2.0	18.3.2019	Analyse économique
3.0	26.3.2019	Analyse du projet existant
4.0	1.4.2019	Analyse technologique
5.0	25.4.2019	Conception et réalisation
6.0	30.4.2019	Tests et évaluations
7.0	5.5.2019	Conclusions
7.1	8.5.2019	Finalisation

TABLE DES MATIÈRES

Document.....	3
1 Métadonnées	3
2 Organisation et liens	3
3 Table des versions.....	3
I. Introduction.....	11
1 Présentation personnelle.....	11
2 Contexte	11
3 Problématique	11
4 Objectif du projet	12
5 Contraintes	12
6 Plan	12
7 Activités	13
7.1 Activités principales.....	13
7.2 Activités optionnelles.....	14
II. Analyse économique	15
8 Tendance du marché et enjeux.....	15
Introduction.....	15
8.1 Stratégie Cybersanté (eHealth) Suisse 2.0.....	15
8.1.1 Loi fédérale sur le dossier électronique du patient	16
8.2 Dossier électronique du patient (DEP).....	17
8.2.1 Architecture eHealth Suisse	18
8.2.2 Fournisseurs de solutions DEP	19
8.3 Coordination internationale	20
Synthèse.....	21
9 Etude de marché.....	22
Introduction.....	22
9.1 Offre actuelle de e-sculape	22
9.2 Motivations	23
9.2.1 Problèmes	23
9.2.2 Besoins	24

9.3	Ciblage de la clientèle	24
9.4	Proposition de valeur	25
9.5	Concurrence et alternatives	27
9.5.1	Systèmes primaires supportant les DEP	27
9.5.2	Services de numérisation	28
9.5.3	Systèmes secondaires de DEP et communautés	28
9.6	Stratégie	29
9.6.1	Modèle économique et positionnement	29
9.6.2	Distribution	30
9.6.3	Forces, faiblesses, opportunités, menaces (SWOT)	31
	Synthèse	32
III.	Analyse du projet existant.....	33
	Introduction	33
10	Implémentation.....	33
10.1	Workflow actuel	33
10.1.1	Réception du document	33
10.1.2	Apposition des codes QR	33
10.1.3	Numérisation	35
10.1.4	Schéma du workflow actuel	37
10.2	Nouveau workflow hypothétique	38
11	Architecture.....	39
11.1	Flux de données	39
11.2	Structure de la base de données	40
12	Données données en entrée.....	41
12.1	PDF/A	41
12.2	OCR	41
12.3	Confidentialité	42
13	Formatage en sortie.....	43
13.1	CDA-CH	43
	Synthèse	43
IV.	Analyse technologique	44
	Introduction	44
14	Analyse de texte	45
14.1	Fuzzy String Matching	45

14.1.1	Hamming	46
14.1.2	Levenshtein	47
14.1.3	Damerau-Levenshtein	47
14.1.4	Jaro	47
14.1.5	Jaro-Winkler	48
14.1.6	Problèmes potentiels	48
14.2	Named Entity Recognition	49
14.2.1	State-of-the-Art NER Models	51
14.2.2	Problèmes potentiels	52
14.3	NLP pour reconnaître le type de document.....	53
15	Analyse d'image.....	54
15.1	Ressources	54
15.2	Document classification.....	54
15.3	Image classification	55
15.3.1	Convolutional Neural Network (CNN)	55
15.3.2	Transfer learning	57
15.3.3	Training Dataset.....	59
15.3.4	Problèmes potentiels	62
15.4	Triplet Network	62
15.4.1	One Shot Learning.....	62
15.4.2	Triplet Loss.....	63
15.5	MorphNet	64
	Synthèse.....	64
V.	Conception.....	65
	Introduction.....	65
16	Workflow et architecture.....	65
16.1	Phase d'entraînement.....	65
16.2	Phase de prédiction.....	67
17	Environnement	68
17.1	Kaggle	68
18	Deep Learning Library.....	70
18.1	Keras	70
VI.	Réalisation	71
	Introduction.....	71

19 Extraction de l'OCR en texte.....	71
20 Transformation PDF en JPG	72
21 Transfer Learning from CNN	73
21.1 Trier les données dans des sets	73
21.2 Transformer les données.....	76
21.3 Créer le modèle de CNN	80
21.3.1 VGG-16.....	81
21.3.2 ResNet-50.....	83
21.3.3 InceptionV3.....	84
21.3.4 Xception.....	84
21.4 Evaluer le modèle.....	84
21.5 Sauver le modèle	86
22 Named Entity Recognition	86
22.1 Modèle existant	86
22.1.1 Français	87
22.1.2 Anglais	88
22.2 Modèle ré-entraîné.....	89
23 Fuzzy String Matching	91
23.1 Levenshtein	92
23.2 Jaro.....	93
Synthèse.....	94
VII. Tests et évaluation.....	95
24 Performances des CNN.....	95
24.1 Fiabilité	95
24.1.1 Visualisation de la fiabilité	95
24.2 Temps d'entraînement total.....	96
24.3 Améliorer la fiabilité du CNN.....	97
24.3.1 Evaluation selon la forme des courbes.....	97
24.3.2 Evolution de la fiabilité sur CNN-16	98
24.3.3 Learning Rate	100
24.3.4 Trainable Weights	101
24.3.5 Dropout.....	102
24.4 Performances avec dataset "balanced"	102
25 Performances NER	104

26 Performances FSM	104
26.1 Evaluation des distances	105
Synthèse.....	106
VIII. Conclusions	107
27 Conclusion du projet.....	107
27.1 Validation des objectifs	107
27.2 Problèmes rencontrés et solutions apportées	108
27.2.1 Données de e-sculape non adaptées	108
27.2.2 RVL-CDIP	108
27.2.3 Transfer Learning avec Keras.....	108
27.2.4 Analyse de texte	108
27.2.5 Mauvaise compréhension du DEP	108
27.2.6 Organisation.....	109
27.3 Perspectives futures.....	109
27.3.1 Regrouper les documents	109
27.3.2 Triplet Loss.....	109
27.3.3 NLP pour reconnaître le type de document.....	109
27.3.4 NER	109
27.3.5 RVL-CDIP	109
27.3.6 Intégration de la solution.....	110
27.4 Remerciements	110
27.5 Conclusion du projet	110
27.6 Conclusion personnelle	111
28 Conclusion du document.....	112
28.1 Licences	112
28.2 Déclaration d'honneur	112
IX. Références.....	113
X. Glossaire	115
XI. Table des figures.....	116
XII. Annexes	120
1 Planning.....	120
2 Kaggle Jupyter Notebook (CNN-16)	121

I. INTRODUCTION

1 Présentation personnelle

Moi, Patrick Audriaz, me suis vu attribuer un projet pour mon sixième semestre en filière télécommunication, orientation internet et communication, à l'école d'ingénierie et d'architecture de Fribourg. Il vise à développer plusieurs compétences, dont la gestion de projet, les présentations orales et la rédaction de rapport. Un projet concret sera donc réalisé au cours de ce semestre par mes soins avec l'assistance et la supervision de deux professeurs responsables : Monsieur Andreas Fischer et Monsieur Nicolas Schroeter.

2 Contexte

La solution qui sera créée pour ce projet a pour but de venir, à terme, se greffer sur une application existante développée en WinDev¹ par l'entreprise d'informatique médicale "e-sculape²" situé à Granges-Paccot (Fribourg). Cette application créée préalablement a pour but de simplifier la migration d'une documentation papier à une documentation informatisé pour les cabinets médicaux. L'application a pour vocation de faciliter leur travail de numérisation de documents médicaux afin de respecter le standard eHealth³ (Stratégie Cybersanté Suisse 2.0) de la Confédération Suisse, dont notamment, la diffusion du dossier électronique (DEP) du patient.

Elle supporte déjà un système de classement au moyen de codes QR associées à un type de document et des informations liées à celui-ci (nom du médecin, du patient...) et un système de reconnaissance optique de caractères (OCR). Le processus actuel d'apposition des codes QR est manuel (collage d'autocollants), le but est donc d'autonomiser cette partie.

3 Problématique

Le processus actuel utilisé par l'entreprise e-sculape pour scanner et classer les documents médicaux est très chronophage car, comme indiqué dans le chapitre 2 ci-dessus, il requiert la main de l'homme afin de fonctionner (apposition de codes QR).

¹ <https://www.pcsoft.fr/windev/index.html>

² <https://www.e-sculape.ch/fr>

³ <https://www.e-health-suisse.ch/fr/page-daccueil.html>

4 Objectif du projet

Il faut fournir à e-sculape un prototype de système de reconnaissance de documents médicaux préalablement scannés. Il faut reconnaître de manière autonome le type du document scanné et extraire les informations qui y sont liées : le nom du patient, sa date de naissance et le nom du médecin.

E-sculape demande également une étude de marché, de concurrence et de modèles économique afin de rendre le produit commercialisable.

5 Contraintes

En travaillant avec des documents médicaux de patients, la question de la confidentialité est très importante afin de garantir le secret médical. Nous utiliserons donc pour ce projet uniquement des documents anonymisés de patients décédés.

6 Plan

Ce rapport a pour but de refléter les étapes de réflexion ainsi que de documenter le travail effectué au cours du semestre et mettre en avant les résultats obtenus. Le document sera structuré ainsi :

Après la rédaction d'un cahier des charges afin de mettre en évidence les aspects sur lesquels mon projet de semestre va se focaliser, un planning sera réalisé afin de garantir une planification optimale du temps imparti. Une fois ceci validé, une analyse économique détaillée sera faite afin de mieux visualiser la place de ce projet dans le monde réel. Une analyse technologique sera également réalisée sur les différentes technologies qui seront employées. Ces analyses nous permettront d'enchainer sur une partie conception en ayant toutes les clés en main pour réaliser un travail cohérent, réaliste, conforme au cahier des charges et servant de base pour la suite du projet. Le travail se terminera sur la réalisation de la solution ainsi que son test pour en valider la fiabilité.

7 Activités

7.1 Activités principales

1. Analyse économique

- 1.1. Analyse du marché pour comprendre les enjeux et les objectifs de la stratégie eHealth
 - 1.2. Analyse de la concurrence pour mettre en évidence des alternatives
 - 1.3. Etude de modèles économiques afin de rendre le produit commercialisable
- **Livrable :** Analyse permettant au mandant de positionner son produit sur le marché et d'en évaluer le potentiel.

2. Analyse du projet existant

- 2.1. Compréhension du fonctionnement générale de la méthode de scanning déjà développée
 - 2.2. Analyse des documents données en input (qualité et quantité)
 - 2.3. Définition d'un formatage des données en output
- **Livrable :** Analyse succincte du projet existant afin de soutenir l'analyse technologique et orienter les choix.

3. Analyse technologique

- 3.1. Recherche et réflexion sur les différents moyens d'arriver à l'objectif de manière optimale
 - 3.2. Description et documentation de ces moyens afin d'en choisir un adapté au projet
 - Machine Learning ?
 - String Matching ?
 - Autre... ?
- **Livrable :** Choix de technologie permettant de répondre à l'objectif du projet

4. Conception et réalisation d'une solution pour reconnaître de manière autonome le type d'un document scanné ainsi que les informations qui y sont liées et les regrouper.
 - 4.1. Modélisation de la solution
 - 4.2. Programmation de l'algorithme
 - 4.3. Tests et évaluation du fonctionnement et de la précision de la solution

→ **Livrable** : Algorithme répondant à l'objectif.

7.2 Activités optionnelles

1. Intégration de la solution dans l'application déjà développée
 - 1.1. Connexion et intégration à l'application WinDev existante (output en CDA-CH)
 - 1.2. Connexion à la base de données HSFQS

→ **Livrable** : Une application intégrant de manière harmonieuse notre solution et étant prêt à être commercialisée

II. ANALYSE ÉCONOMIQUE

8 Tendance du marché et enjeux

Introduction

Une analyse économique fait partie du cahier des charges, le chapitre "Tendance du marché et enjeux" a pour but de répondre à la question :

- Quels sont les enjeux et objectifs de la stratégie cybersanté de la Confédération Suisse ainsi que de la loi "Loi fédérale sur le dossier électronique du patient" et quels sont leurs influences et comment elles motivent ce projet ?

8.1 Stratégie Cybersanté (eHealth) Suisse 2.0

*"Objectifs et mesures de la Confédération et des cantons pour diffuser le dossier électronique du patient et coordonner la numérisation autour du dossier électronique du patient."*¹



La "Stratégie Cybersanté (eHealth) Suisse 2.0" a pour objectif de remplacer la "Stratégie Cybersanté (eHealth) Suisse" du 27 juin 2007. C'est une mesure élaborée par les cantons et la Confédération qui vise à accompagner les différents acteurs vers leur transition numérique et accomplir plusieurs objectifs d'ici **2022** dont :

- **Encourager la numérisation** : que le dossier électronique du patient (DEP) soit instauré et diffusé globalement au sein du système de santé Suisse.
- **Harmoniser et coordonner la numérisation** : permettre l'utilisation multiple de données et d'infrastructures.
- **Habiliter à la numérisation** : rendre les personnes responsables de trainer les données digitales des patients en ayant conscience des risques.

¹ <https://www.bag.admin.ch/bag/fr/home/strategie-und-politik/nationalegesundheitsstrategien/strategie-ehealth-schweiz.html>

La numérisation vise à rendre le système de santé Suisse plus efficace, sûr et de meilleure qualité. Elle permet une interconnexion et un échange de données facilitées entre les institutions, les professionnels de la santé et les patients grâce au canal numérique, le tout en garantissant un contrôle et un suivi des accès.

8.1.1 Loi fédérale sur le dossier électronique du patient

Tout ceci est mis en place afin de respecter les délais imposés par la "**Loi fédérale sur le dossier électronique du patient¹**" (**LDEP**). Cette loi définit les objectifs, les aspects organisationnels, techniques, sécuritaires et les prescriptions concernant le **dossier électronique du patient (DEP)**.

Toutes les institutions médicales ne sont cependant pas soumises à la LDEP. "*Le délai pour les hôpitaux est fixé au 15 avril 2020 et celui pour les maisons de naissance et les EMS au 15 avril 2022*"². Les cabinets médicaux privés et les médecins privés ne sont donc pas obligés d'intégrer le DEP et peuvent garder un système traditionnel. Ils peuvent cependant tout de même, et sont encouragés, à mettre en place un système de DEP qui s'adresseraient à certaines catégories professionnelles plus spécifiques comme des pharmacies ou des médecins établis en cabinet.

Cette décision de ne pas traiter tous les acteurs de la santé en Suisse de la même manière a été prise à cause du faible taux d'informatisation des cabinets privés (seulement 44,8% des médecins de famille et 36,4% des spécialistes selon une étude³) et parce que ces acteurs spécifiques n'ont pas forcément le temps de se plier à cette loi, problème que les hôpitaux n'ont pas grâce à un effectif plus grand. Cela gâche un peu le potentiel de cette initiative.

(1) (2)

¹ <https://www.admin.ch/opc/fr/classified-compilation/20111795/index.html>

² https://www.e-health-suisse.ch/fileadmin/user_upload/Dokumente/2017/F/170804_Wer_muss_ein_EPD_anbieten_f.pdf

³ <https://www.letemps.ch/economie/dossier-electronique-patient-cest-un-changement-fondamental-dapproche>

8.2 Dossier électronique du patient (DEP)

"C'est un instrument pour les patients et les professionnels de la santé visant à renforcer la qualité des traitements médicaux, à améliorer les processus de traitement des patients, à accroître la sécurité des patients, à augmenter l'efficience du système de santé et à favoriser la compétence des patients en matière de santé".¹



Figure 8.1 : Logo du DEP²

Le DEP est concrètement un dossier ou une archive virtuelle accessible sur toute plateforme (web et mobile). **Il constitue la documentation interne d'un professionnel de la santé et contient des documents personnels importants relatifs à la santé d'un patient** comme des vaccins, des radiographies, des ordonnances, des rapports d'entrée et de sortie, des directives anticipées, des prescriptions, des examens... Chaque professionnel de la santé enregistre les documents pertinents dans le DEP de leur patient. Le DEP est ouvrable gratuitement pour toute personne résident en Suisse.

Son contrôle est entre les mains du patient, il choisit les informations qui y sont partagées et les personnes qui y ont accès (sauf en cas d'urgence ou l'accès est libéré). Il permet de rassembler de manière centralisée et uniformisée toutes informations pouvant être utiles en cas de consultation médicale, que ce soit pour une chirurgie lourde ou un simple contrôle chez le dentiste. Le DEP n'est cependant pas un système d'archivage et les assurances-maladies n'y ont pas accès.

¹ <https://www.e-health-suisse.ch/fr/politique-droit/bases-juridiques/loi-federale-ldep.html>

² <https://www.patientendossier.ch/fr>

8.2.1 Architecture eHealth Suisse

Il ne faut pas confondre le dossier médical électronique (système primaire) et le dossier électronique du patient¹² (DEP ou système secondaire). Ce sont deux notions différentes :

Système primaire ou dossier médical électronique

Il constitue la documentation interne du professionnel de la santé. Ce sont les données des patients qui sont stockés par le médecin et ses collaborateurs dans le système informatique propre à l'établissement. (ERP).

Système secondaire ou DEP

Comme expliqué plus haut, le DEP permet de mettre en réseau et à disposition des professionnels de la santé les informations pertinentes et importantes pour le traitement d'un patient et nécessitant son consentement pour y accéder. Il est utilisé pour faciliter l'échange d'informations dans notre société où les gens sont de plus en plus nomades.

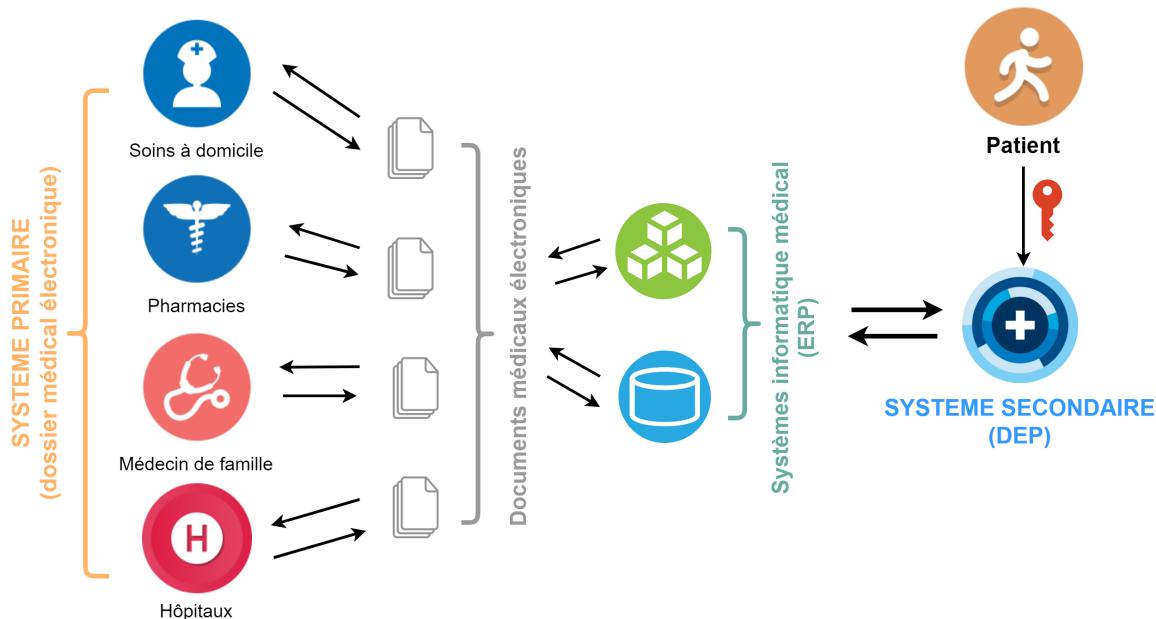


Figure 8.2 : Schéma de l'architecture pour la mise en place du DEP

¹ https://www.e-health-suisse.ch/fileadmin/user_upload/Dokumente/2015/F/151208_fiche_dinformation_difference_dossier_medical_electrique_dossier_electronique_patient_F.pdf

² <https://www.e-health-suisse.ch/fr/mise-en-oeuvre-communautes/mise-en-oeuvre/questions-et-reponses.html>

Comme nous pouvons l'observer dans les schémas ci-dessus, le processus d'échange d'informations du côté professionnels est le suivant :

Les divers acteurs de la santé saisissent les différents documents et informations des patients de manière numérique dans leur système informatique propre. Une partie ou la totalité de ces informations est mise à disposition sur le DEP en fonction des besoins du patient. Chaque professionnel de la santé peut ensuite, selon les besoins, aller consulter les documents pertinents auxquels il a les droits d'accès.

Le DEP est une gestion décentralisée des données qui renvoie simplement aux données pertinentes pour les soins des patients qui stockées dans les systèmes informatiques respectifs des professionnels de la santé. C'est pour cela que les documents numérisés se doivent de respecter des normes très précises de formatage afin d'être inter-compatibles, cela est assuré grâce au format CDA-CH qui sera plus détaillé par la suite.

8.2.2 Fournisseurs de solutions DEP

La confédération Suisse laisse libre choix aux établissements et cabinets médicaux de leurs fournisseurs de DEP. Elle ne prévoit pas de solution globale mais seulement des directives et normes techniques¹ à suivre ainsi qu'un descriptif d'architecture afin d'assurer une interopérabilité. La confédération a organisé plusieurs "Projectathon" afin de faire des test pratiques pour des implémentations de DEP.

Plusieurs organisations privées (Post CH, Afga HealthCare, NEXUS Schweiz, SwissSign Group...) se sont rencontrées et se sont affrontées afin de tester et comparer leur système en prévision de l'entrée en vigueur du DEP et en vue d'une potentiel Certification DEP². Ils fournissent des plateformes (multiplateformes) de documentation, de visualisation et de gestion de DEP pour les patients et les professionnels. "*MonDossierMedical.ch*"³ en est un exemple déjà implanté (par la Poste.).

¹ <https://www.e-health-suisse.ch/fr/technique-semantique/interoparabilite-technique/normes-techniques.html>

² <https://www.e-health-suisse.ch/fr/mise-en-oeuvre-communautes/communautes-dep/certification-dep.html>

³ <https://www.mondossiermedical.ch/faq>

Malgré l'absence de solution globale, plusieurs régions se sont regroupées en "**communautés de référence¹**" et ont commencé à créer des solutions et offres relatives au DEP auxquels les hôpitaux et cabinets pourront s'affilier. Par exemple :

- **Cara. (Réalisation par La Poste)²** : Genève, Valais, Vaud, Fribourg et Jura
- **Verein eHealth Zentralschweiz³** : Lucerne, Obwald, Nidwald
- **XAD/axsana SA/Swisscom Health⁴** : Zurich, Schaffhouse, Saint-Gall, Berne

8.3 Coordination internationale

eHealth s'inspire de près des divers projets de cybersanté qui prennent forme au niveau Européen, ceci dans le but de ne pas créer un système entièrement fermé.

La Commission Européenne a cependant indiqué récemment que la Suisse ne pourra plus collaborer au sein des organes de coordination européen de cybersanté. Cette décision s'explique car la Suisse a décidé de ne pas suivre une directive proposée en 2013 visant à permettre de se faire soigner partout dans l'UE et de se faire rembourser les coûts de prestation.⁵

Plusieurs réseaux se sont formés au niveau européen afin de proposer des solutions de cybersanté globales :

- **eHealth Network⁶** : l'organe décisionnel stratégique de l'Union européenne pour tout ce qui touche au domaine "eHealth". La Suisse n'en fait plus parti.
- **Connecting Europe Facility-Programme⁷** : ouvre toutes les infrastructures de l'économie d'échange numérique de données dans l'Union européenne. La Suisse n'en fait plus parti.

¹ <https://www.e-health-suisse.ch/fr/mise-en-oeuvre-communautes/communautes-dep/communautes-en-cours-de-constitution.html>

² <http://www.cara.ch/>

³ <https://www.ehzs.ch/>

⁴ <http://www.axsana.ch/>

⁵ <https://www.e-health-suisse.ch/fr/politique-droit/bases-strategiques/coordination-internationale.html>

⁶ https://ec.europa.eu/health/ehealth/overview_en

⁷ <https://ec.europa.eu/digital-single-market/connecting-europe-facility>

- **Integrating the Healthcare Enterprise¹** : a pour objectif d'améliorer l'échange d'informations de santé entre les différents systèmes informatiques. La Suisse en fait partie.

Synthèse

Comme nous avons pu le constater avec l'état des lieux ci-dessus, la mise en place de la loi LDEP est quelque peu chaotique. Cela représente des changements importants dans le système de fonctionnement des instituts de santé Suisse et les délais imposés sont courts.

Les enjeux sont cependant clairs, les intentions louables et les patients et médecins pourront en retirer de nombreux avantages.

¹ <https://www.ihe.net/>

9 Etude de marché

Introduction

Une étude de marché fait partie du cahier des charges, ce chapitre a pour but de répondre aux questions :

- Que propose actuellement e-sculape et comment leur offre va évoluer pour répondre à cette nouvelle demande ?
- Quels sont les acteurs déjà actifs sur ce nouveau marché ? Quels sont les solutions qui sont déjà proposées, qui seront proposées ou quels sont les alternatives pour les cabinets médicaux ?
- Quel doit être le positionnement de e-sculape et quel modèle économique leur permettra d'en faire un produit commercialisable et rentable ?

9.1 Offre actuelle de e-sculape

Comme indiqué dans le chapitre 2 ci-dessus, l'application existante a déjà mis en place un système permettant de simplifier la migration des documents papiers vers une version numérique. Cela répondait initialement au besoin qu'avaient les médecins de se conformer aux nouvelles dispositions de la "LDEP" en leur proposant un service de scannage de leurs documents physiques.

E-sculape propose comme service principal un ERP (logiciel qui permet de gérer l'ensemble des processus opérationnels d'une entreprise en intégrant plusieurs fonctions de gestion) de la santé pour les cabinets. Leur service d'ERP a comme modèle économique une location mensuelle avec un serveur installé chez le client. Le service de scanning et de DEP vient donc étoffer leur offre en étant proposé comme complément indépendant à leur ERP.

Leur solution de scanning vient remplir un vrai besoin. En effet, les cabinets médicaux perdent trop de temps et d'argent à numériser leurs dossiers médicaux. En développant cette solution pour simplifier la numérisation des documents, e-sculape propose un service très alléchant pour ceux-ci qui n'ont pas forcément les mêmes effectifs et moyens que les hôpitaux.

L'implémentations actuelle est cependant pas encore optimisée dû au fait que le scanning requiert l'apposition d'autocollants par un humain afin de trier et labelliser les documents et les lier à des entrées dans leur ERP. Ce processus est couteux en temps pour e-sculape ou pour un cabinet et ils souhaitent l'automatiser afin d'augmenter leur rendement et rendre leur offre plus attractive pour les cabinets médicaux en diminuant les coûts et les délais.

(3)

9.2 Motivations

Ces notions de "problèmes et besoins" vont nous permettre de prouver que le produit qui va être développé est innovant, nécessaire et utile à une certaine démographie et qu'il a le potentiel d'être viable. Cela nous permettra de modéliser notre solution. Il est nécessaire de les identifier car elles viendront aiguiller nos décisions futures lors de l'analyse afin de positionner notre produit sur un marché et face à une potentielle concurrence. Un nouveau produit se doit de répondre à un problème et de le résoudre sans quoi il risque de ne pas trouver son public. Pas de demande, pas de ventes.

9.2.1 Problèmes

Cabinets médicaux

Malgré qu'ils ne soient pas obligés, la Confédération conseille fortement aux cabinets médicaux de se conformer à la LDEP d'ici 2022. Les cabinets médicaux sont cependant souvent très occupés et n'ont ni l'effectif, ni le temps d'effectuer ces tâches de numérisation.

Les médecins partant à la retraite ou en cas de reprise d'activité, les cabinets ont besoin d'un moyen de stockage d'information numériques dans un objectif d'archivage.

E-sculape

La solution actuelle de e-sculape est très ingénieuse et répond à un vrai besoin mais n'est pas aboutie. Elle n'est dans l'état pas viable parce qu'elle est très chronophage comme indiqué dans le chapitre 3 ci-dessus.

9.2.2 Besoins

Cabinets médicaux

Il faut un service de scanning fiable, rapide et abordable permettant aux cabinets médicaux d'assurer leur virage vers le numérique.

E-sculape

Il faut à e-sculape une extension de leur logiciel actuel afin d'avoir un produit répondant pleinement aux attentes des cabinets médicaux. Un système de scanning quasi-autonome et fiable éliminant la nécessité d'apposer des codes QR sur les documents avant de les scanner.

9.3 Ciblage de la clientèle

Le ciblage de la clientèle est essentiel dans une analyse afin de concorder l'offre et la demande. En effet, si on espère toucher tout le monde, on risque de toucher personne. Il faut être au courant de sa principale clientèle, ou "coeur de cible" afin de définir et d'adapter son positionnement sur le marché. Le ciblage de la clientèle est surtout utile pour élaborer une stratégie marketing mais un produit sera tout de même intimement lié à la démographie qu'il souhaite atteindre. Il se doit d'être en concordance avec les exigences attendues qui varient grandement d'une population à l'autre. Des professionnels auront d'autres exigences au niveau du ton et des canaux utilisés et de la sécurité des données comparé à des particuliers par exemple. Il faut faire ce travail d'identification de cibles afin de les comprendre et de s'y adapter.

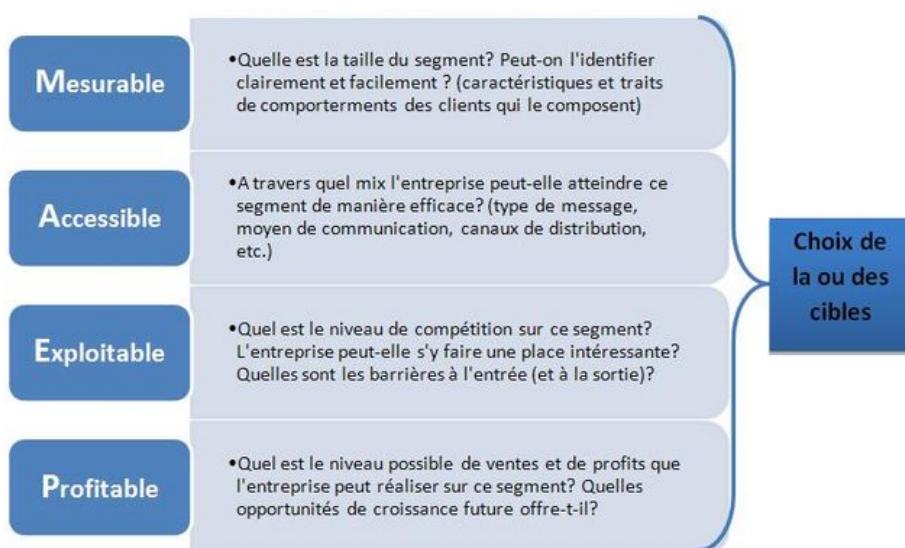


Figure 9.1¹ : Critères d'aide au choix d'un cible

¹ <http://www.marketing-professionnel.fr/wp-content/uploads/2011/10/criteres-choix-cible.jpg>

Deux cibles principales sont à relever pour ce projet, ce sont des professionnels dans le domaine de la santé en Suisse. Elles remplissent les critères cités dans la Figure 9.1 ci-dessus

- Cabinets médicaux qui veulent passer du papier au numérique.
- Médecins qui reprennent l'activité d'un médecin qui partent à la retraite et qui veulent numériser leurs archives, soit pour faciliter le stockage, soit pour passer à un système numérique et dématérialisé.

Maintenant que les cibles sont définies, nous pourrons adapter notre produit afin de répondre au mieux à leurs attentes.

9.4 Proposition de valeur

La proposition de valeur est une promesse que nous nous engageons à tenir envers nos clients. C'est ce qui fait qu'ils choisissent notre produit plutôt que celui d'un concurrent. Son but est de montrer le côté unique de notre solution, d'en prêcher les avantages et de prouver qu'il résout un problème. La proposition de valeur est également là pour donner une idée claire au client des services que nous proposons et des bénéfices qu'ils pourront en tirer. Nous pouvons la modéliser grâce au canvas de proposition de valeur.

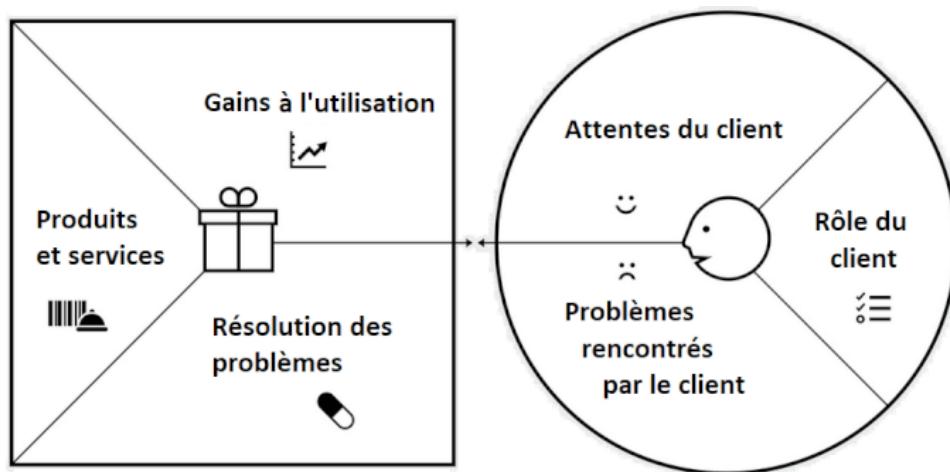


Figure 9.2¹ : Canvas de proposition de valeur

¹ <https://www.creerentreprise.fr/wp-content/uploads/2018/07/canevas-proposition-valeur-1.png>

Les problèmes et attentes (besoin) du client ont déjà été formalisés ci-dessus, il nous reste à formuler la vraie proposition de valeur de notre produit sous forme de canvas :

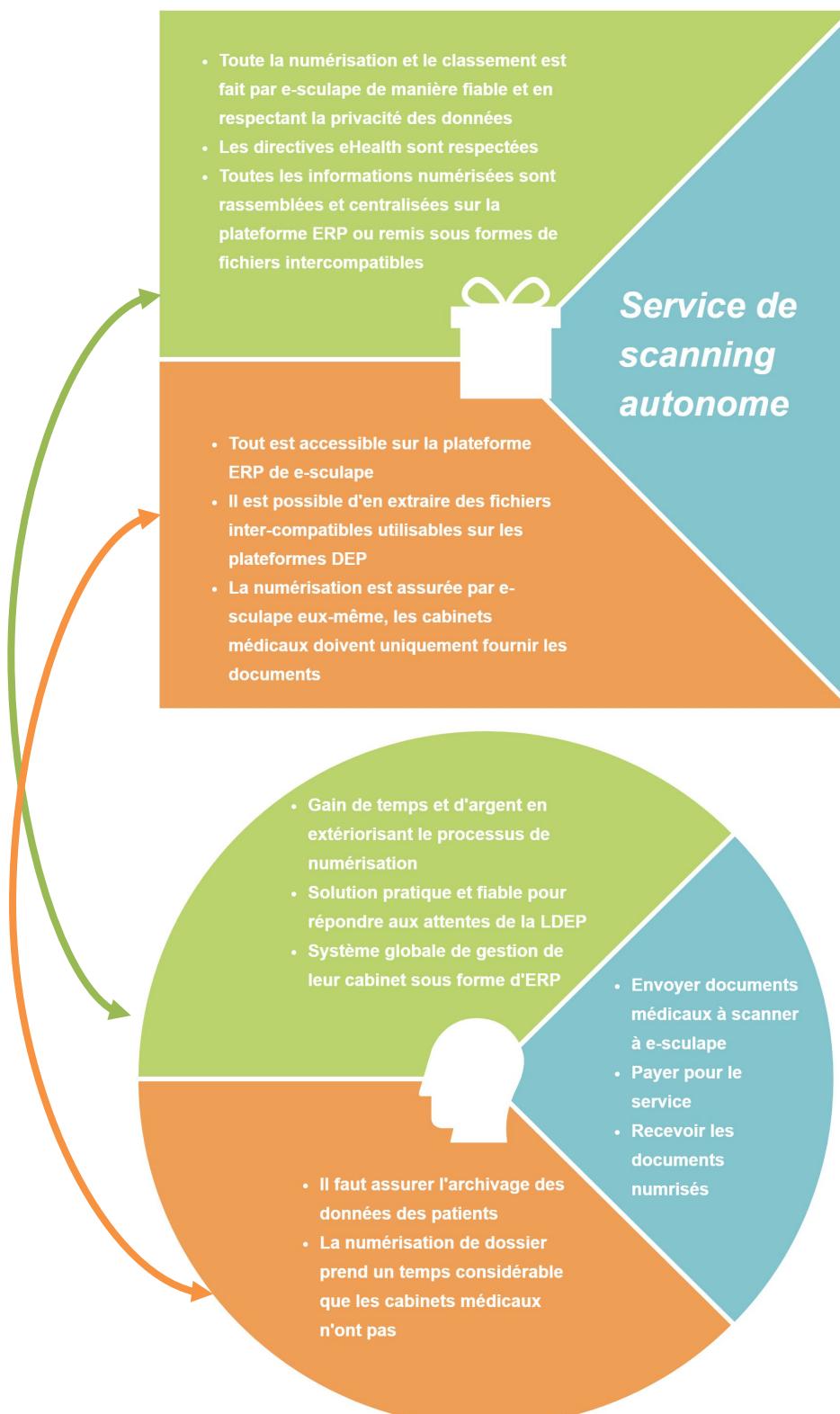


Figure 9.3 : Modélisation du canvas de proposition de valeur pour notre projet (e-sculape en haut, cabinets médicaux en bas)

9.5 Concurrence et alternatives

Il est essentiel de connaitre la position de ce projet dans le marché, de voir les différentes variantes et solutions trouvées dans celui-ci et surtout la pertinence de l'option mise en place dans ce projet.

Selon mes recherches, plusieurs solutions sont déjà envisageables (dont les communautés de référence mentionnées au point 8.2.2 ci-dessus), il faut être conscient que la plupart sont encore en état de tests et seront prêtes que d'ici 2020 pour répondre aux attentes de la Confédération Suisse. Il y a plusieurs entreprises qui offrent des ERP dans le domaine de la santé et qui souhaitent proposer des solutions de DEP intégré. Il y a également des entreprises qui offrent uniquement des services de numérisation de document avancé et d'autre qui offrent seulement une plateforme pour les DEP.

Aucune entreprise ne propose cependant de solution d'ERP de la santé complète avec numérisation autonome de documents et format compatible DEP comme e-sculape souhaite le faire.

9.5.1 Systèmes primaires supportant les DEP

LOGICARE¹

Entreprise privé, système informatique hospitalier, numérisation et solution DEP (certification en cours)



Amétiq medical (DocBox) ²

Entreprise privé, ERP de la santé, solution DEP (certification en cours)



¹ https://www.logicare.ch/index.php/123/ServicesL%C3%B6sungen/Business_Application/Beratungsl%C3%B6sungen/eHealth-Prozess-Check

² <https://www.ametiq.com/de>

AVINTIS eHealth-Gateway¹

Entreprise privé, ERP de la santé, solution DEP (certification en cours)

**9.5.2 Services de numérisation****Arcplace²**

Entreprise privée, numérisation, classification et extraction de données

**9.5.3 Systèmes secondaires de DEP et communautés****Post E-Health³**

En cours de développement, assure la réalisation technique pour la communauté de référence "Cara". Solution de DEP. S'appuie sur "Siemens Healthineers eHealth Solutions".



¹ <https://www.avintis.com/fr/ehealth-center>

² <https://www.arcplace.ch/en/offer/solutions/scanning/>

³ <https://www.post.ch/fr/entreprises/index-thematique/solutions-sectorielles/solution-sectorielle-sante/dossier-electronique-du-patient-dep>

Axsana AG / XAD / Swisscom Health¹

Communauté de référence pour Zurich et Berne. Solution DEP gérée par Swisscom Health. C'est une fusion de plusieurs sociétés dans le but de créer le plus grand réseau de cybersanté Suisse.



MyEPD²

Communauté, solution DEP qui sera utilisé par la ville de Bâle.



9.6 Stratégie

9.6.1 Modèle économique et positionnement

Le service de scannage autonome que nous souhaitons proposer répond à une réelle demande comme je l'ai démontré. Le but d'une entreprise est de multiplier ses clients afin d'augmenter les profits. Il serait donc judicieux de faire de ce service de scanning le "**produit d'appel**" de e-sculape. C'est-à-dire un service sur lequel est pratiqué un prix avantageux dans le but d'amener des nouveaux clients dans leur écosystème (ERP de la santé).

J'ai identifié plusieurs cas de figures des potentiels client, nous pourrons en tirer plusieurs "formules" de modèle économique :

¹ <http://www.axsana.ch/>

² <https://www.myepd.ch/>

Client actuel de l'ERP de e-sculape qui souhaite numériser ses documents

Pour cette démographie-là, le service de scannage sera seulement proposé et facturé comme service supplémentaire à l'abonnement qu'ils paient déjà pour profiter de l'ERP de e-sculape.

Cabinets médicaux recherchant une solution d'ERP et se dirigeant vers e-sculape car ils proposent ce service de scanning

C'est pour cette démographie là que le "produit d'appel" sera le plus attractif afin de gagner un nouveau marché grâce à la solution proposée. Nous l'utiliserons pour s'en servir comme porte d'entrée au service d'ERP déjà proposé par e-sculape.

Nous pouvons nous permettre de le proposer à un tarif préférentiel car les cabinets médicaux paieront déjà le service d'ERP de la santé et l'automatisation de la tâche de scannage de document la rend très peu couteuse une fois implémenté et fonctionnelle.

Médecin qui reprend l'activité d'un médecin qui part à la retraite et qui cherche une solution de numérisation et de dématérialisation des archives existantes.

C'est un cas un peu particulier, il faut proposer à ce genre de client un forfait avec à la carte une numérisation de ses documents sous formes de fichiers inter-compatibles avec les normes DEP qu'il pourra utiliser comme archive ou pour assurer une reprise d'activité sans être dépendant de l'ERP.

9.6.2 Distribution

Nous nous positionnons dans le domaine de la santé, et plus précisément pour les cabinets médicaux. La vente se fera en canal direct, en contact direct avec les cabinets médicaux afin de répondre au mieux à leurs attentes et assurer un suivi. C'est nécessaire car le domaine de la santé doit répondre à de nombreuses contraintes.

Le scannage pourrait être un service qui s'effectuera dans les locaux de e-sculape, aucun besoin pour les médecins de se déplacer, ils auront seulement à leur envoyer leurs documents. Nous aurons ensuite juste à les leur faire parvenir, soit au travers de l'ERP, soit au travers de fichiers inter-compatibles avec les normes DEP.

9.6.3 Forces, faiblesses, opportunités, menaces (SWOT)

Une analyse SWOT permet de mettre facilement en évidence les forces, faiblesses, opportunités et menaces d'un projet afin de mieux identifier la stratégie nous permettant d'atteindre les objectifs souhaités.

C'est un outil utilisé durant les analyses préliminaires pour aider la suite du processus¹.

- **Opportunité** : circonstances et occasions qui créer un environnement favorable au projet, une innovation technologique, une évolution du marché...
- **Menace** : problème potentiel posé par l'environnement ou le marché qui pourrait empêcher notre projet de trouver son public
- **Force** : Capacités et compétences de l'investigateur du projet lui procurant un avantage ou distinction net sur un marché qui offre un avantage concurrentiel
- **Faiblesse** : Potentiel handicap dans un domaine

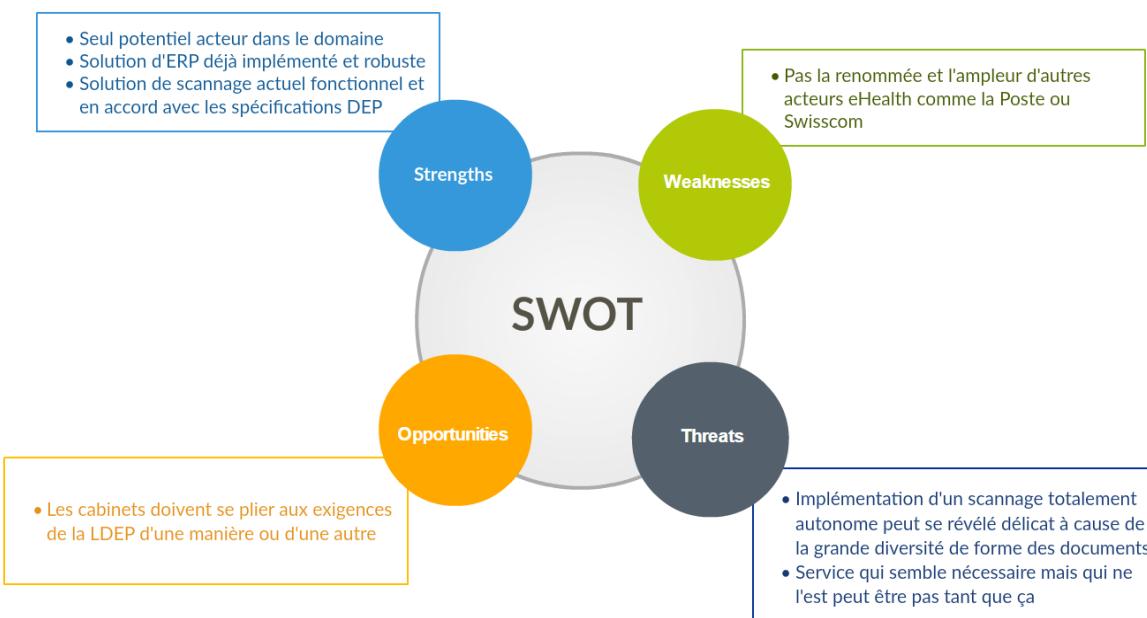


Figure 9.4 : Analyse SWOT

¹ https://cyberlearn.he-sso.ch/pluginfile.php/144036/mod_resource/content/0/Theorie_analyse_SWOT.pdf

Synthèse

Nous pouvons aisément constater que ce projet est nécessaire, bénéfique et utile. Nous avons étudié de nombreux aspects afin de confirmer ces dires : une analyse du contexte du domaine de la cybersanté en Suisse, une analyse de la concurrence, des clients, du positionnement... Nous avons modélisé les différentes opportunités et menaces que nous risquons de rencontrer en commercialisant ce projet mais nous avons maintenant, avec cette analyse, toutes les clés en main pour assurer sa réussite commerciale.

Nous pouvons donc continuer ce projet en étant confiant du potentiel de notre produit.

III. ANALYSE DU PROJET EXISTANT

Introduction

Je vais analyser le projet existant afin de m'aider à modéliser ma solution. Comprendre comment il fonctionne et quel est son architecture afin de proposer une solution cohérente. Les points suivants vont être documentés :

- Documentation du processus de numérisation actuel.
- Modélisation et explication de l'architecture actuelle et du flux des données.
- Analyse des documents données en entrée au logiciel
- Normes et contraintes que doivent respecter les données obtenues en sortie.

E-sculape ont déjà passablement bien documenté leur système actuel. C'est pourquoi je ne vais pas trop entrer dans les détails et analyser seulement les points qui sont essentiels à mon projet. (3)

10 Implémentation

10.1 Workflow actuel

Le processus de numérisation imaginé par e-sculape et qui est pour le moment en place est le suivant :

10.1.1 Réception du document

1. E-sculape possède dans leur ERP une base de données ou une liste Excel de tous leurs clients (médecins) ainsi que de tous leurs patients.
2. Le dossier complet du patient est envoyé par un médecin, il est reçu par un opérateur et il lui assigne le nom du médecin et tripe les documents par catégorie et par ordre chronologique.

10.1.2 Apposition des codes QR

3. L'opérateur indique le nom du patient du dossier, sa date de naissance et le matching est fait avec une entrée dans la base de données. Il le crée s'il n'existe pas dans la base de données.

4. Il indique ensuite les types de documents contenus dans le dossier sur l'application dédiée

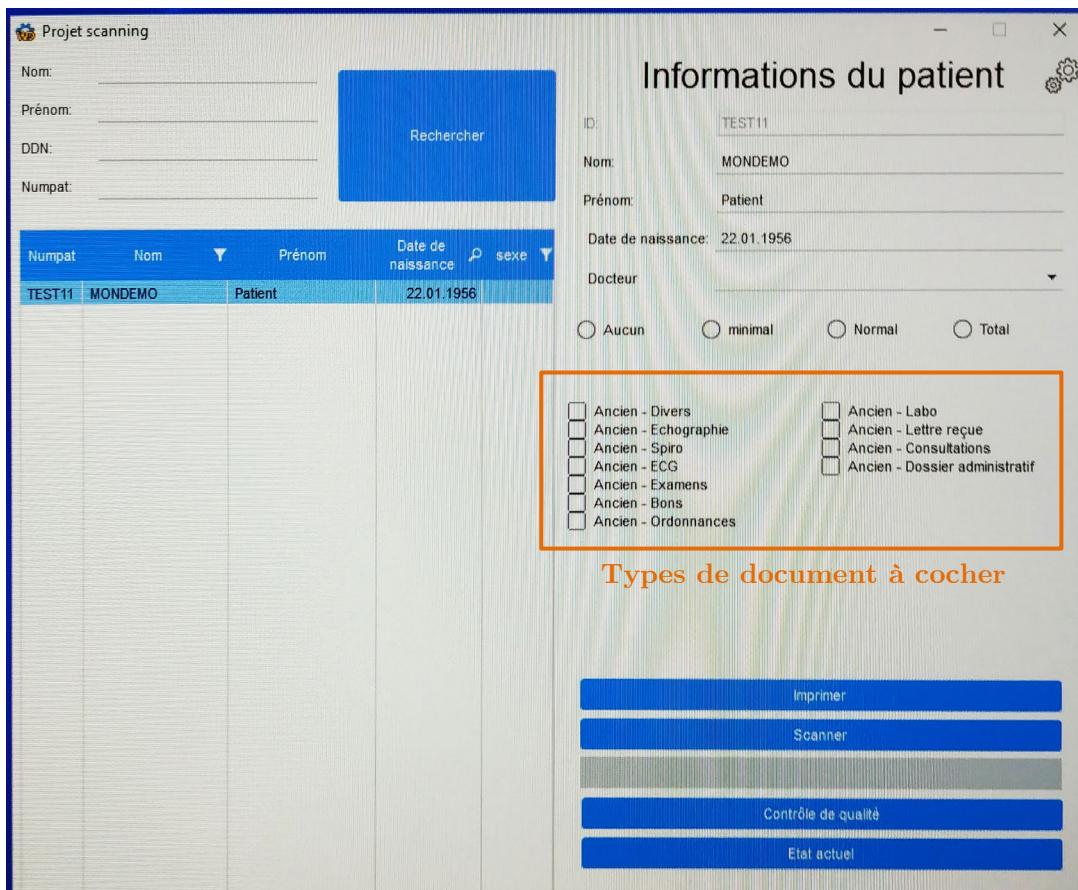


Figure 10.1 : Logiciel pour entrer les informations du patient et imprimer les codes QR

5. Le logiciel génère ensuite et imprime des autocollants de code QR (fonctionnement détaillé plus loin).



Figure 10.2 : Exemples d'autocollants de code QR générées

6. Les codes QR sont collés sur les documents. S'il y a plusieurs pages pour un type de document, seule la première page reçoit un code QR.



Figure 10.3 : Codes QR collé sur les documents en fonction de leur type

10.1.3 Numérisation

7. Les documents sont ensuite placés (regroupés par type de document) tous dans un appareil de numérisation.
8. Le logiciel scan les documents sous forme PDF/A et effectue de la reconnaissance de caractères (OCR avec *Canon CapturePerfect*¹). Un documents PDF est généré par feuille scannée.
9. Le logiciel reconnaît le code QR et rassemble tous les documents précédant un nouveau code QR sous le type de document indiqué par le code QR précédent.

¹ <https://fr.canon.ch/scanners/document-scanners/capture-perfect/>

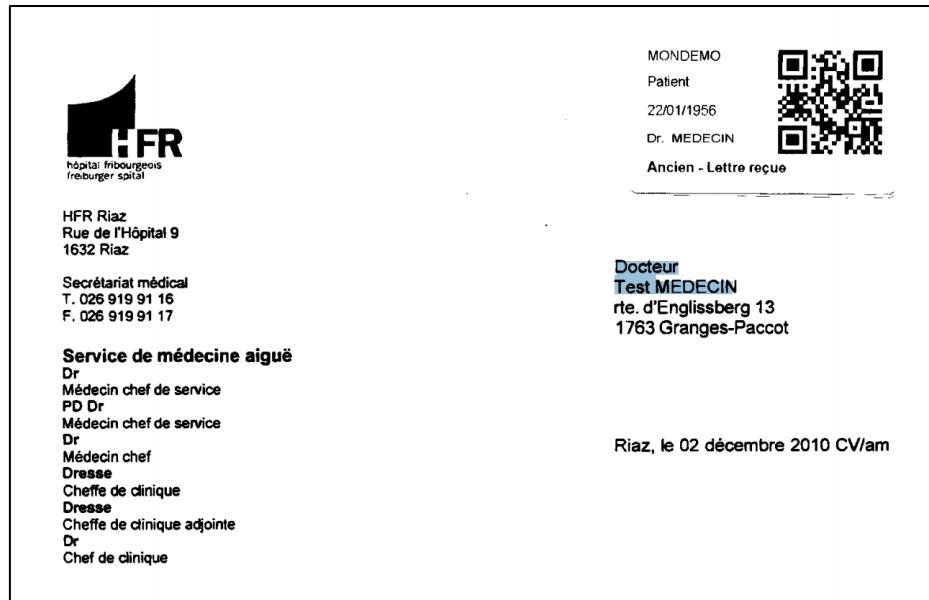


Figure 10.4 : Exemple de fichier PDF sortie avec OCR

10. Génération du CDA-CH (détaillé plus bas), création d'un fichier XML contenant les informations extraites du code QR et le PDF/A encapsulé en base64.

```
<?xml version="1.0" encoding="UTF-8"?>
<ClinicalDocument xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <realmCode code="FR" />
  <typeId root="POCD_HD000040" root="2.16.840.1.113883.1.3" />
  <templateId root="2.16.840.1.113883.2.8.2.1" />
  <templateId root="1.3.6.1.4.1.19376.1.2.20" />
  <id root="1.125.11.555.98.0.10" />
  <code code="11528-7" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LN" displayName="TEST">
    </code>
  <title>Ancien - Ordonnances</title>
  <effectiveTime value="20180822105650+0200" />
  <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25" displayName="Normal" />
  <languageCode code="fr-FR" />
  <setId root="1.125.11.555.98.1.401442" />
  <versionNumber value="1" />
  <recordTarget>
    <patientRole>
      <id extension="G3R104203889" root="1.125.11.555.101.33.0" />
      <addr>
        <streetName>rte. d'Englisberg 13</streetName>
        <city>GRANGES-PACCOT</city>
        <postalCode>1763</postalCode>
      </addr>
    </patientRole>
  </recordTarget>

```

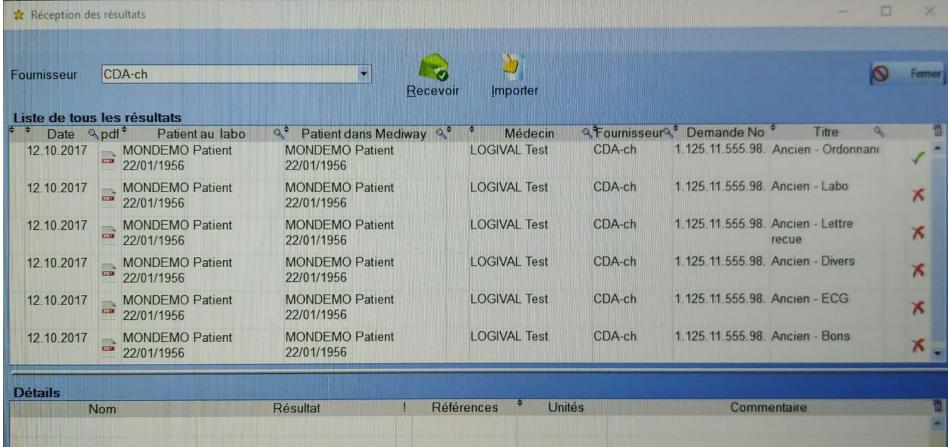
Figure 10.5 : En-tête de l'XML (CDA-CH) généré

```
<component>
  <nonXMLBody>
    <text mediaType="application/pdf"
      representation="B64">JVBERi0xLjcNCiWhs8XXDQoxIDA...Gz9Y...lcyAyIDA...UiAvVHlwZS9DYX...hbgG...
    QgMi9La...Rw...yA0IDA...UiAgMTAgMCBSIF0vVHlwZS9QY...ld1cz4
    +DQp1bmRvYmoNCjMgMCBVYmoNCjw8L0NyZWF0a...w9uRGF0ZShEo...jIwMTkwMzA1MTQyNTA5KS9DcmVhdG9yKFBErml1bSk
    +DQp1bmRvYmoNCjQgMCBVYmoNCjw8L0NvbnR1bnRz...wyA1IDA...UiAgNiAwIFIgIDcgMCBSIF0vT...vkaWF...Cb3hbIDAgMC...
    Vzb3V...yY2VzPDwvRm9udDw8L0YzIDggMCBSID4+L1...y2NTZX...rl1BERi9UZ...xh0L01tY...dlQi9JbWF...nZUMvSw1hZ2VJ...
    +L1R5cGUvUGFnZT4+DQp1bmRvYmoNCjUgMCBvYmoNCjw8L0ZpbHr1ci9GbGF0ZUR1Y29kZ...S9MZW5ndGggMz...u+PnN0cm...
    +JDHA1iA9xj0jx9+MCAGkjb6MNCmVuZHN0cmVhbQKZ...W5kb2JqD...o2IDA...gb2JqD...o8PC9Ga...d0Z...IvRmxhdgVEZ...W...
  </nonXMLBody>

```

Figure 10.6 : Encapsulation du PDF en base64 dans le XML

11. Validation, contrôle de qualité du scannage par l'opérateur.
12. Upload sur la base de données de l'ERP fourni par e-sculape.



The screenshot shows a software interface titled 'Réception des résultats'. At the top, there is a dropdown menu 'Fournisseur' set to 'CDA-ch', and three buttons: 'Recevoir', 'Importer', and 'Ficher'. Below this is a table titled 'Liste de tous les résultats' with columns: Date, pdf, Patient au labo, Patient dans Mediway, Médecin, Fournisseur, Demande No, Titre, and several red checkmark icons. The table lists six entries from 12.10.2017, all from 'MONDEMO Patient' with birth date '22/01/1956'. The 'Titre' column contains descriptions like 'Ancien - Ordonnance', 'Ancien - Labo', 'Ancien - Lettre recue', 'Ancien - Divers', 'Ancien - ECG', and 'Ancien - Bons'. At the bottom, there is a 'Détails' section with columns: Nom, Résultat, Références, Unités, and Commentaire.

Figure 10.7 : Documents scannés et données associées disponibles sur l'ERP

10.1.4 Schéma du workflow actuel

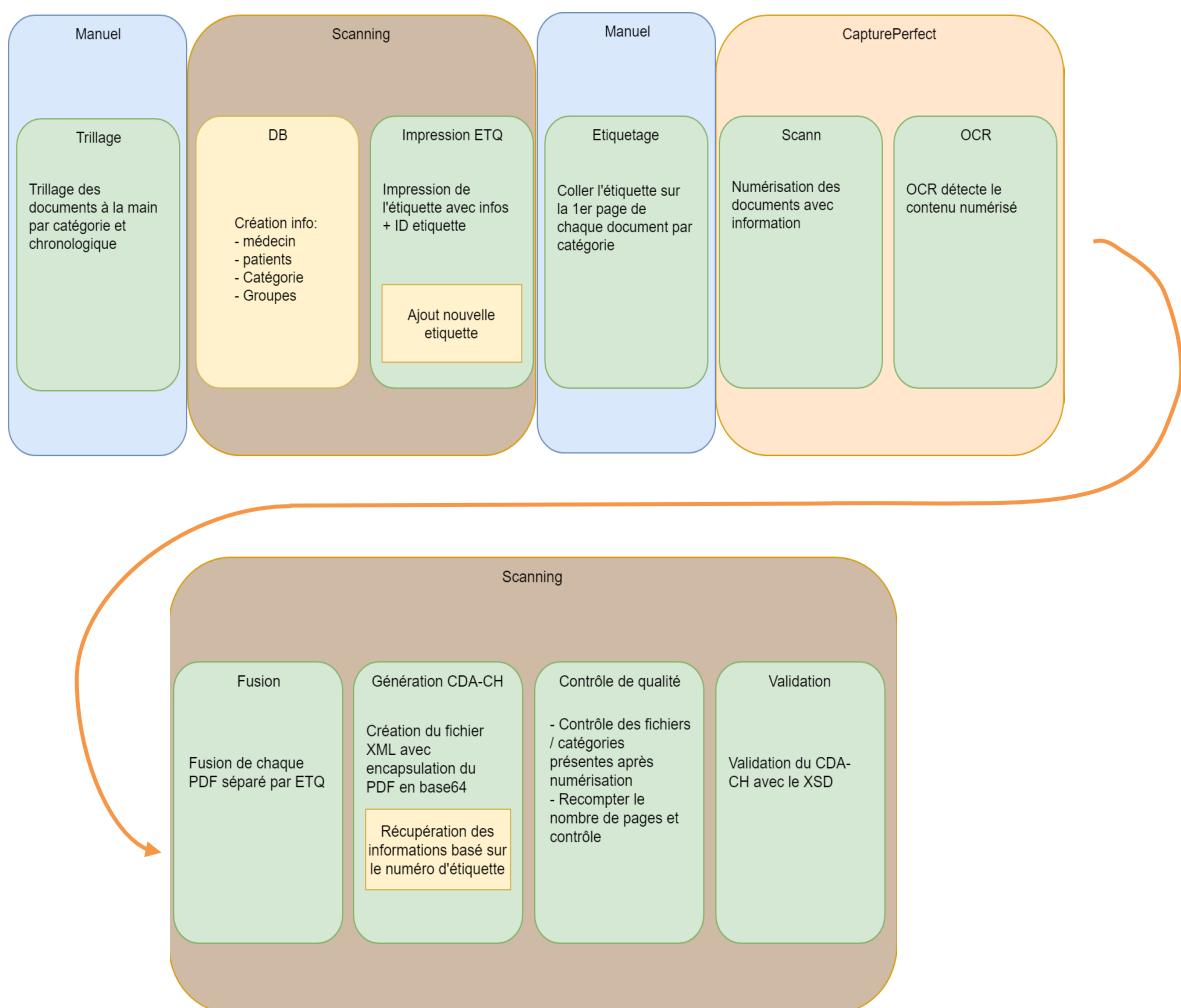


Figure 10.8 : Schéma du workflow actuel

10.2 Nouveau workflow hypothétique

Comme maintenant indiqué de nombreuses fois, le but est de s'affranchir de la nécessité de devoir classer les documents à la main. Pour ce faire, le nouveau workflow imaginé est le suivant, il montre ou l'application créée pour ce projet viendra s'intégrer (en rouge) et les étapes qu'elle vise à remplacer (en gris). Nous verrons plus loin, dans la conception du la solution, dans quelle mesure cela est faisable en tenant compte des diverses contraintes.

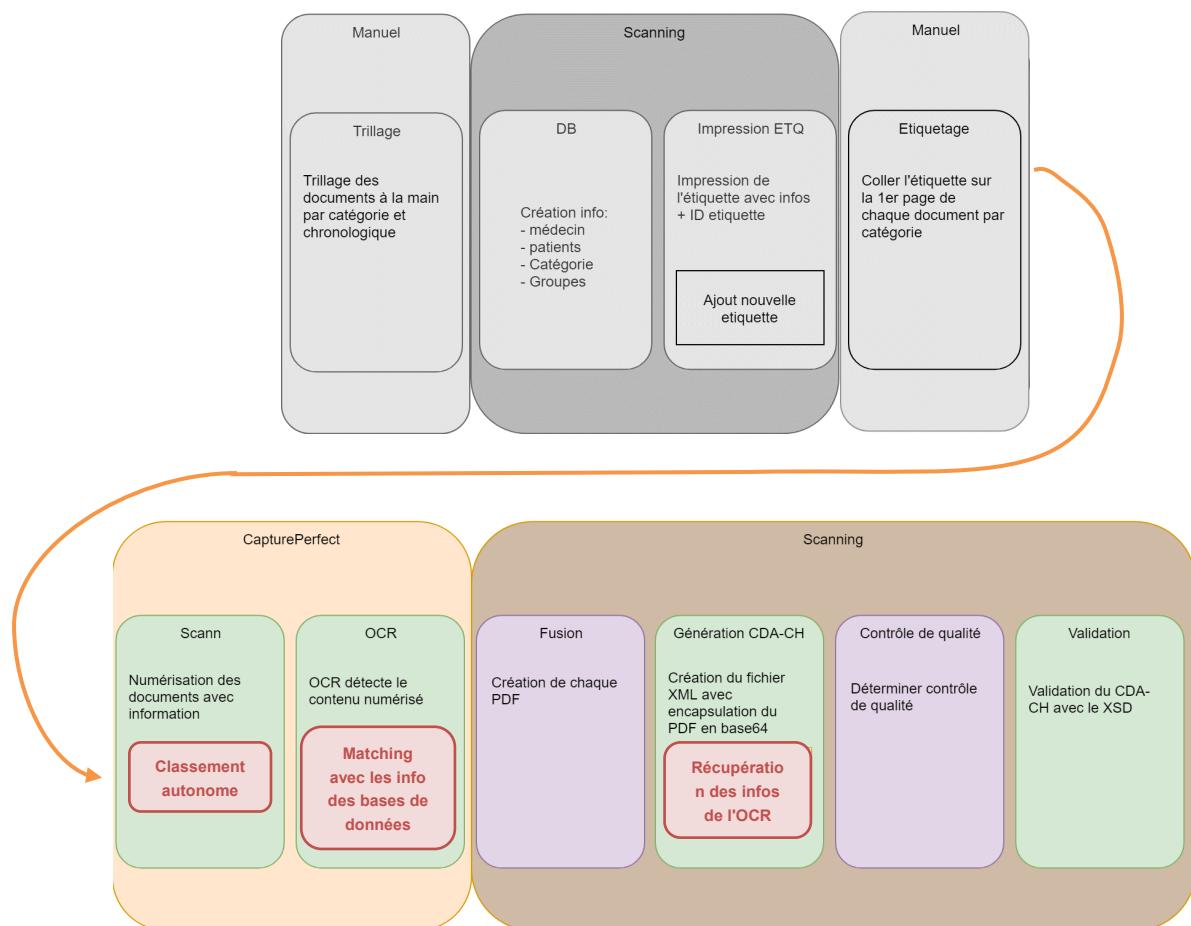


Figure 10.9 : Schématisation du nouveau workflow hypothétique

Toute la partie manuelle du scanning est donc omise en faveur d'une solution automatisée. On commence directement à l'étape du scan et la classification autonome s'y fait. Le workflow continue ensuite et relie les informations extraites grâce à l'OCR aux document classifiés. Le reste de la chaîne ne change pas. Un contrôle de qualité manuel reste nécessaire mais néanmoins allégé afin de garantir la fiabilité du scannage dans un milieu aussi important que la santé.

11 Architecture

11.1 Flux de données

Le schéma ci-dessous nous permet de visualiser tous les "acteurs" de la solution actuelle et de comprendre leurs différentes interactions :

0. Documents à scanner envoyés
1. Les cabinets sont enregistrés chez e-sculape (si nouveau client)
2. Les infos des patients sont ajoutées à l'ERP (si nouveau client)
3. Le service de scanning utilise ces infos
4. L'opérateur se connecte au service
5. L'opérateur trie les dossiers des patients à scanner
6. L'opérateur indique les informations des codes QR
7. Les codes QR correspondants aux documents sont imprimés
8. Les étiquettes sont collées sur les documents
9. Les documents sont scannés en PDF
10. OCR et génération des PDF
11. Génération du CDA-CH

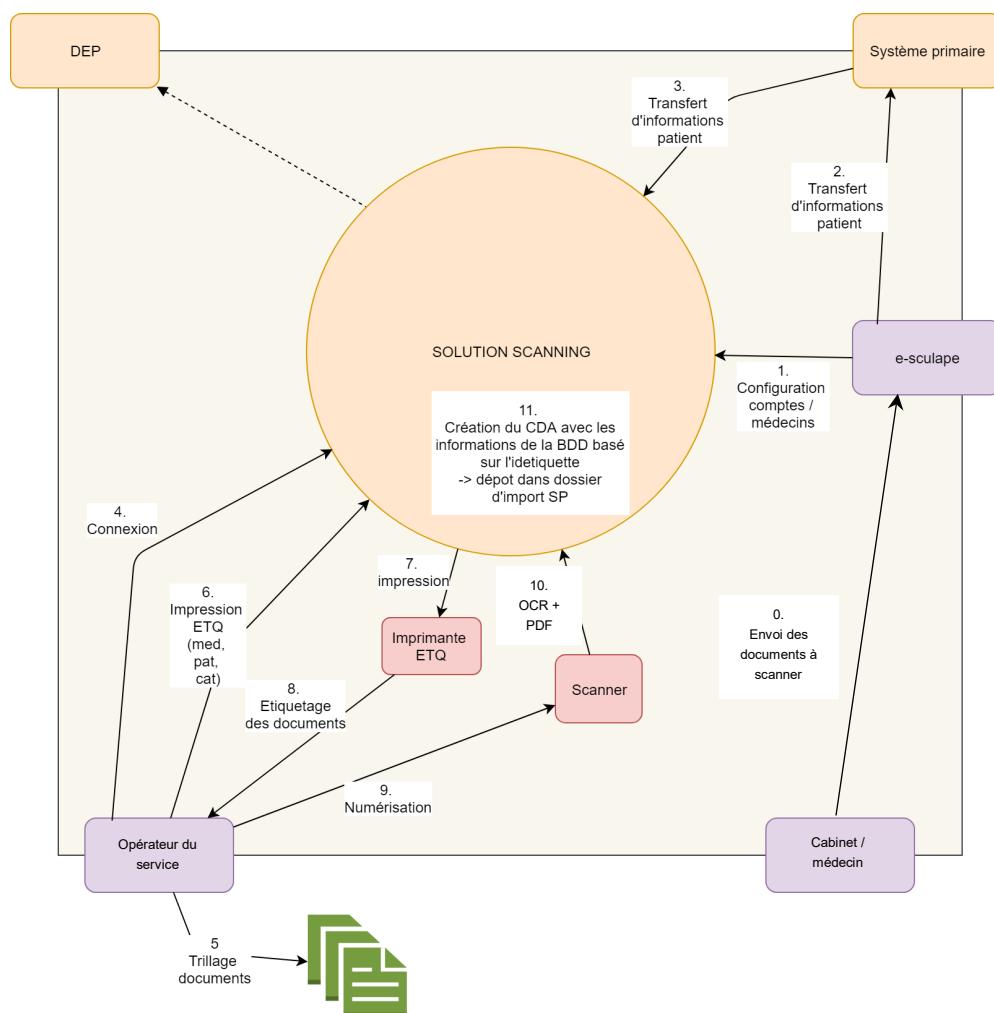


Figure 11.1 : Diagramme du flux des données actuel

11.2 Structure de la base de données

Les données sont structurées d'une manière particulière dans la base de données afin de permettre aux code QR de contenir toutes les informations concernant un document (type, patient, date de naissance, médecin...) sans les compliquer et pour ainsi faciliter la reconnaissance de ceux-ci grâce à la conversion du PDF en bitmap.

Le code QR (sur l'étiquette) ne contient qu'un ID, un unique identifiant l'étiquette (IDEtiquette).

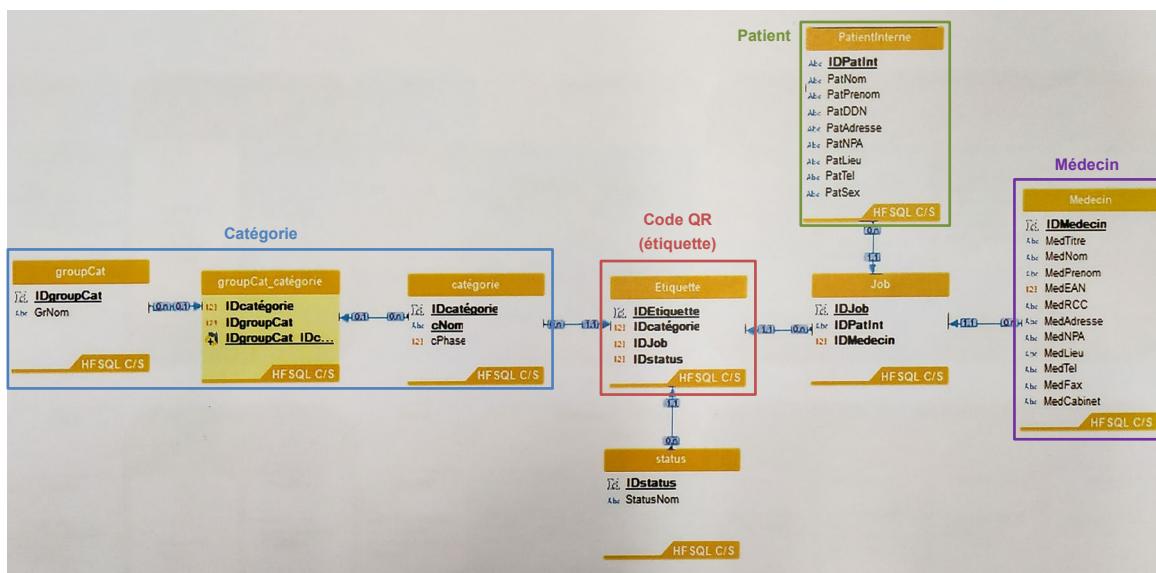


Figure 11.2 : Diagramme de classe de la base de données

Pour le scannage autonome, il faudra donc si possible respecter la structure de cette base de données afin de faciliter l'implémentation de la solution dans l'architecture déjà mise en place chez e-sculape. Les bases de données exportées sont au format XLS (Microsoft Excel). Nous allons les convertir en JSON par soucis de praticité.

```
{
  "Nº_Enr.": [1],
  "IDMedecin": [1],
  "MedTitre": ["Dr. "],
  "MedNom": ["AUDRIAZ"],
  "MedPrenom": ["Patrick"],
  "MedEAN": [7601000222111],
  "MedRCC": ["A123456"],
  "MedAdresse": ["rte. d'Englisberg 13"],
  "MedNPA": [1763],
  "MedLieu": ["Granges-Paccot"],
  "MedTel": ["026 550 05 80"],
  "MedFax": ["026 550 05 81"],
  "MedCabinet": ["Cabinet du Dr. Audriaz Patrick"]
}
```

Figure 11.3 : Exemple d'entrée de la base de données (médecin) exportée en JSON

12 Données données en entrée

Il est ici question des données qui résultent du scannage (PDF et OCR) que nous pourrons utiliser pour donner en entrée à notre solution autonome.

12.1 PDF/A

Les documents sont scannés par l'appareil Canon en version PDF/A¹. C'est un format similaire au PDF traditionnel, à la différence qu'il a été conçu spécialement pour l'archivage à long terme et la conservation de documents sous forme électronique en maximisant la compatibilité.

Le scanner crée un document PDF par document scanné, ils seront rassemblés par type plus tard. Grâce aux codes QR pour le moment et grâce à l'automatisation dans le futur.

12.2 OCR

Le logiciel utilisé pour reconnaître générer les PDF et reconnaître les caractères des documents scannés est "Capture Perfect" de Canon². C'est une solution grand-public qui n'est pas des plus fiable comme le montre l'exemple ci-dessous. Il faut donc mettre en place un mécanisme (string matching) pour pallier les erreurs potentielles de ce système d'OCR.

On peut aisément extraire les données d'OCR récoltés par le logiciel sous format TXT afin de les utiliser dans notre solution.



Figure 12.1 : Exemple d'erreur avec le PDF à gauche et l'OCR à droite

```
> pdftotext in.pdf out.txt
```

Figure 12.2 : Pour récupérer l'OCR d'un PDF sous forme de texte

¹ <https://en.wikipedia.org/wiki/PDF/A>

² <https://fr.canon.ch/scanners/document-scanners/capture-perfect/>

12.3 Confidentialité

En travaillant avec des documents médicaux de patients, la question de la confidentialité est très importante afin de garantir le secret médical. C'est pourquoi e-sculape ne peut nous fournir que des documents de patients décédés et anonymisés. Cela est fait en utilisant des alias pour les noms et en masquant au moyen de gros rectangles noirs les informations considérées sensibles.

Ce genre de masquage très intrusif pose un problème à un système de reconnaissance autonome car il se base sur la structure (image analysis) plutôt que sur le contenu du document (text analysis). Ces rectangles noirs seront donc, au moment de l'entraînement, enregistrés comme étant des propriétés intrinsèques au type du document analysé. L'anonymisation des documents prenant également beaucoup de temps, il nous est impossible de disposer d'un dataset assez grand pour convenablement entraîner un algorithme à la reconnaissance de documents.

Genre de prestations dispensées / prestations de la / des organisation-s d'aide et de soins à domicile :		Remarques du médecin :
Soins de base 1/1an. Contrôle santé 1/1an. Administration médic. 1/1an. Instruction - Conseil 1/1an.		
Infirmier/ère référent/e: _____ Sceau/signature : 		Médecin: _____ Sceau/signature : 
Date : <u>28.XY.2010.</u>		Date : <u>10.07.14</u> No RCC

Figure 12.3 : Exemple d'anonymisation et de masquage

13 Formatage en sortie

13.1 CDA-CH

Comme mentionné dans le chapitre 8.2.1 ci-dessus, il faut une norme permettant de structurer les données, les uniformiser, les unifier pour les rendre inter-compatibles à travers tout le système de santé Suisse. Cette norme se nomme : CDA-CH. Les documents CDA-CH sont codés en XML.

Je ne vais pas m'attarder sur ce sujet, e-sculape ont déjà réglé son implémentation dans l'application existante.

(4)

```

<?xml version="1.0" encoding="UTF-8"?>
<?xmlstylesheet type="text/xsl" href="vhltg-cda-v3.xsl"?>
<ClinicalDocument
    xmlns="urn:hl7-org:v3"
    xmlns:voc="urn:hl7-org:v3/voc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:hl7-org:v3 CDA.xsd">

    <!--
    ****
    En-tête CDA
    ****
    -->
    <component>
        <structuredBody>
            <!--
            ****
            Corps CDA, contient plusieurs sections
            ****
            -->
            <component>
                <section>
                    <title>titre</title>
                    <text>
```

Figure 13.1 : Exemple de structure XML de document CDA-CH

Synthèse

Nous pouvons constater que nous avons beaucoup de contraintes à respecter afin d'implémenter la solution dans l'application existante. Le projet n'est cependant, pour le moment, qu'un test de faisabilité.

IV. ANALYSE TECHNOLOGIQUE

Introduction

Nous allons, pour ce projet, utiliser le système d'OCR déjà intégré à l'installation actuelle. Nous avons deux tâches distinctes à effectuer pour la numérisation de ces documents médicaux :

- Reconnaître le type du document (ordonnance, formulaire de sortie, prescriptions, examens...)
- Extraire les métadonnée (nom du médecin, nom du patient, date de naissance du patient, institution)

Pour découvrir ces informations de manière autonome, plusieurs solutions peuvent être envisagés :

- Analyse de texte : on analyse le texte donné par l'OCR et on en extrait les informations désirées.
- Analyse d'image : on analyse l'image, sa structure pour faire des prédictions.
- Reconnaissance mixte : analyse de texte pour les métadonnées et analyse d'image pour le type de document.

Nous allons voir dans ce chapitre un éventail de ces différentes méthodes et de quelle manière elles peuvent être mises en œuvre afin de parvenir à une solution fonctionnelle et optimale. Le but est de faire un choix de technologies à mettre en pratique dans la réalisation.

14 Analyse de texte

Nous pouvons analyser le texte des documents scannés pour en extraire des informations utiles. Plusieurs méthodes existent :

14.1 Fuzzy String Matching

Une idée est d'utiliser le texte extrait par l'OCR et de le comparer à une base de données ou un dictionnaire afin de trouver des correspondances ou des "match" avec des éléments contenus dans celle-ci comme des noms, prénoms ou des types de document.

Cette idée de faire correspondre exactement un texte à un autre s'appelle le "String Matching". Nous avons vu dans le chapitre 12.2 ci-dessus que l'OCR produit beaucoup d'erreurs et il nous faut un moyen de les corriger si nous souhaitons les exploiter.

Le "Fuzzy String Matching¹" entre alors en jeu. Il est utile lorsque nous souhaitons comparer des chaînes de caractères qui ne sont pas totalement semblables mais plus ou moins ou approximativement similaires et que nous souhaitons mesurer leurs similitudes en pourcentage.

Il compare chaque chaîne de caractère et mesure leur similitudes grâce à des opérations primaires comme l'insertion, la suppression et la substitution. Cette comparaison nous donne comme résultat une nombre nommé "**Edit Distance**" qui "compte" le nombre minimum de ces opérations pour transformer une chaîne de caractère en une autre. Le but est d'arriver à des matchs malgré les différences entre les chaînes de caractère.

- insertion: *cot* → *coat*
- deletion: *coat* → *cot*
- substitution: *coat* → *cost*

Figure 14.1 : Exemple des opérations primaires

¹ https://en.wikipedia.org/wiki/Approximate_string_matching

Dans notre projet, une solution de ce type serait exécutée ainsi :

1. Connaître en avance le nom du médecin à qui appartient le dossier médical
 2. Avoir un fichier texte contenant toutes les chaînes de caractères extraites de l'OCR.
- Chaque chaîne de caractère de ce document est appelée "**String**"
3. Avoir une chaîne de caractère de la base de données du médecin (nom du patient, type de document) à comparer avec le fichier texte. Elle est appelée modèle ou "**Pattern**".
 4. Appliquer l'algorithme de Fuzzy String Matching sur chaque paires String/Pattern
 5. Extraire le Pattern qui a obtenu le plus haut pourcentage de correspondance.

Plusieurs algorithmes existent afin d'effectuer cette tâche, je vais parler que de quelques-uns basées sur des caractères et que de ceux qui semblent être le plus adapté à notre cas d'utilisation.

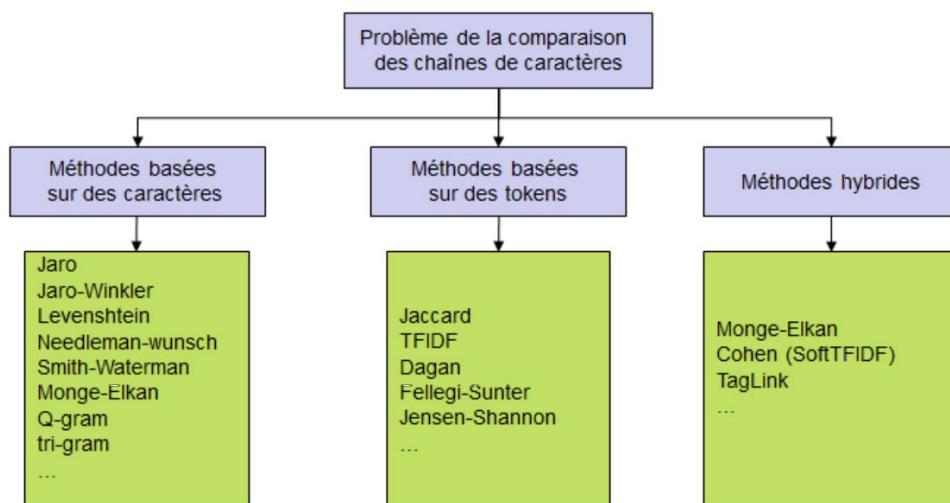


Figure 14.2 : Eventail des différentes méthodes de String Matching (5)

14.1.1 Hamming¹

Algorithme permettant de mesurer la similitude de uniquement deux chaînes de caractère de la même longueur car il ne permet que les **substitutions**.

Algorithme hors-concours car il se peut que l'OCR "oublie" un caractère, donnant des chaînes de caractère qui ne sont pas de la même longueur.

¹ https://en.wikipedia.org/wiki/Hamming_distance

14.1.2 Levenshtein¹

La distance de Levenshtein est une méthode de mesurer la différence entre deux chaînes de caractère. Elle correspond au nombre minimal de caractères qu'il faut **insérer**, **supprimer** ou **remplacer** pour avoir deux chaînes qui match. Chaque opération a un coût ce qui fait que plus il y a d'opération réalisée, plus la distance d'édition est grande, plus la différence entre les deux chaînes de caractère est grande.

Exemple pour passer de "kitten" à "sitting", la distance de Levenshtein est de 3 :

- kitten → sitten (substitutions de "s" pour "k")
- sitten → sittin (substitutions de "i" pour "e")
- sittin → sitting (insertion de "g" à la fin).

14.1.3 Damerau-Levenshtein²

Modification de la distance de Levenshtein prenant en compte, en plus des trois opérations de celle-ci, les **transpositions** qui sont l'inversement de deux caractères adjacents.

Cet algorithme a été créé dans le but de proposer un calcul de la distance d'édition plus fidèle aux erreurs réalisées lors de la frappe à l'ordinateur par un humain. Cet algorithme a donc plutôt pour objectif de réaliser une correction orthographique pour corriger des erreurs de dactylographie plutôt que des erreurs complètement imprévisibles réalisées par l'OCR.

Pour cette raison, sa fiabilité peut être compromise dans notre application et il est donc hors-course.

14.1.4 Jaro³

Mesure de similarité entre deux chaînes de caractère. La distance de Jaro est entre 0 et 1, plus les deux chaînes sont similaires, plus la distance tend vers 1. Cet algorithme utilise le nombre de caractères similaires ainsi que le nombre de transpositions afin de définir une métrique.

¹ https://en.wikipedia.org/wiki/Levenshtein_distance

² https://en.wikipedia.org/wiki/Damerau%E2%80%93Levenshtein_distance

³ https://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance

14.1.5 Jaro-Winkler¹

Modification de l'algorithme de Jaro afin de donner un poids plus important aux erreurs commises en début de chaînes de caractère. Cet algorithme n'est pas pertinent pour notre application car les erreurs de l'OCR peuvent être n'importe où dans la chaîne de caractère.

Nous avons donc deux algorithmes de mesure de différence entre deux chaînes de caractère qui sortent du lot et qui semblent adaptés : **Levenshtein et Jaro**. A voir lors des tests lequel est le plus performant tant en termes de durée d'exécution que de fiabilité dans notre cas d'application.

14.1.6 Problèmes potentiels

Performances

Pour cette approche, nous avons à crawler à travers toute la base de données pour trouver un match. Il faut appliquer l'algorithme de string matching entre toutes les chaînes de caractère extraite par l'OCR du document scanné avec toutes les entrées de la base de données jusqu'à avoir une correspondance. La vitesse du système dépend donc fortement du nombre d'entrées dans la base de données ainsi que du nombre de mots extrait de l'OCR. Exemple :

- Nombre de patients du médecin dans la base de données : 20
- Nombre de mots dans l'OCR : 40
 - Nombre d'opérations : $20 \times 40 = 800$
- Nombre de patients du médecin dans la base de données : 1500
- Nombre de mots dans l'OCR : 300
 - Nombre d'opérations : $1500 \times 300 = 450'000$

Il faudrait donc trouver un moyen de réduire le nombre d'entités à comparer, une potentielle solution est proposée dans le chapitre Named Entity Recognition ci-dessous.

On peut également envisager de se concentrer uniquement sur la partie haute du document qui contient en général les informations importantes.

¹ https://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance

Différencier le médecin du patient

Il est nécessaire pour cette méthode de connaître au moins le nom du médecin à qui appartient le dossier. Si ce n'était pas le cas cela poserait des problèmes pour différencier les médecins et les patients, comment savoir si un médecin n'est pas patient ? Et l'inverse ? Comment ne pas les confondre ? Nous pouvons, en connaissant le nom du médecin, l'éliminer de la liste des informations.

Il peut encore potentiellement avoir un problème si le patient et le médecin ont le même nom ou un nom très similaire.

Nom absent de la base de données

Nous pourrons soit utiliser le Fuzzy String Matching afin de corriger l'OCR ou pour faire des "match" en utilisant la base de données de e-sculape comme dictionnaire. Cette approche ne pourra pas être faite pour les nouveaux clients car ils ne sont pas dans le système. Ils ne bénéficieraient donc pas de cette partie du service.

14.2 Named Entity Recognition¹

Une alternative afin de quand même extraire des informations de l'OCR sans avoir des données préalablement enregistrées sur une base de données pourrait être le Named Entity Recognition.

C'est une sous-tâche du domaine de l'extraction de données dans des documents. Cela consiste à catégoriser des mots dans des classes comme des noms, des entreprises, des lieux, des aliments, des dates... Nous pouvons utiliser cette autre approche pour classifier les métadonnées sans avoir recours à une base de données. En effet, le Named Entity Recognition (NER) s'appuie sur les techniques d'apprentissage supervisé afin de reconnaître et classer les mots dans les différentes catégories. C'est une tâche importante et bien connue dans le domaine du Natural Language Processing (NLP²). Les progrès récents dans le domaine du deep learning ont permis de développer des modèles puissants et fiables (85% à 90% de précision) mais cela nécessite cependant un grand nombre de données d'entraînement.

C'est une méthode qui peut s'avérer être très utile pour notre problème, nous recherchons en effet à retrouver des informations précises à partir de données brutes et à les classifier.

¹ <https://towardsdatascience.com/a-review-of-named-entity-recognition-ner-using-automatic-summarization-of-resumes-5248a75de175>

² https://en.wikipedia.org/wiki/Natural_language_processing

Le but est de "tagger" les différents mots d'un fichier texte non structuré dans différentes catégories prédéfinies, d'extraire les données les plus pertinentes et de filtrer celles qui ne sont pas désirés et qui ne correspondent pas aux catégories.

Dans notre projet, une solution de ce type serait exécutée ainsi :

1. Récupérer les données de l'OCR sous forme de texte
2. Transformer cette liste de chaînes de caractères en tokens
3. Tagger les tokens en fonction de leurs catégories
4. Entrainer le modèle avec les données catégorisées
5. Utiliser le modèle sur nos documents à classifier

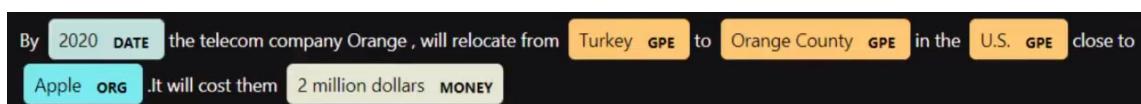


Figure 14.3 : Exemple de résultat de NER

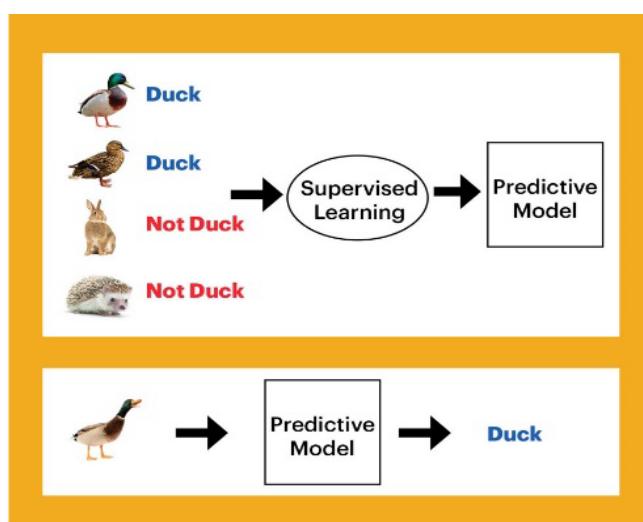


Figure 14.4 : Schéma d'apprentissage supervisé pour de la classification, phase d'entraînement et de prédiction¹

¹ https://2s7gjr373w3x22jf92z99mgm5w-wpengine.netdna-ssl.com/wp-content/uploads/2018/09/WD_2.jpg

14.2.1 State-of-the-Art NER Models¹

SpaCy NER Model²

SpaCy est une librairie Python gratuite et open-source spécialisée dans le NER. Elle est réputée pour ses grandes performances, sa bonne documentation et ses prédictions précises.

Il est possible d'utiliser leurs modèles pré-entraînés, disponibles également en français, ou d'entrainer notre propre modèle afin qu'il réponde spécifiquement à nos attentes.

TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.

Figure 14.5 : Exemples d'entités reconnues par le modèle pré-entraîné de spaCy³

```
import spacy
nlp = spacy.load('en')
to_analyze = ('Hello Code & Supply, '
              'my name is Josh and tonight '
              'we\'re in Pittsburgh')
doc = nlp(to_analyze)
ents = [(x.text, x.label_)
         for x in doc.ents]
print(ents)
ents
```

Josh	PERSON
tonight	TIME
Pittsburgh	GPE

Figure 14.6 : Exemple d'utilisation de spaCy qui ressort les informations intéressantes d'une phrase

¹ <https://towardsdatascience.com/a-review-of-named-entity-recognition-ner-using-automatic-summarization-of-resumes-5248a75de175>

² <https://spacy.io/>

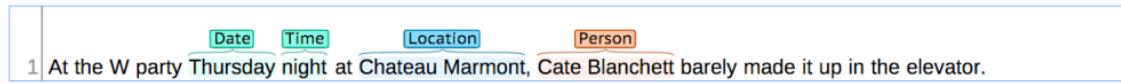
³ <https://spacy.io/api/annotation#named-entities>

Stanford Named Entity Recognizer¹

Ce modèle est implémenté en Java, il propose également un modèle pré-entraîné capable de reconnaître des entités comme des organisations/entreprises, des personnes et des lieux. Il est plus tourné vers l'entraînement de modèles personnalisés avec nos propres dataset de données étiquetées que spaCy.

Le modèle Stanford NER ne supporte, par défaut, pas le français. Il faudrait donc le ré-entraîner pour atteindre nos objectifs.

Named Entity Recognition:



Basic Dependencies:

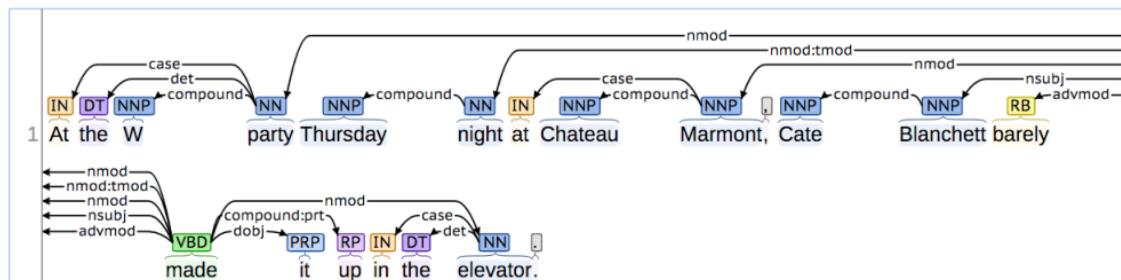


Figure 14.7 : NER par le Stanford Named Entity Recognizer

14.2.2 Problèmes potentiels

Avoir un dataset permettant d'entraîner le modèle

Si le modèle pré-entraîné de, par exemple, spaCy ne convient pas à notre cas d'application, il faudra le ré-entraîner. C'est un processus qui peut être complexe et prendre beaucoup de temps. Il faut rassembler une grosse quantité de données pertinentes et les classer en respectant la syntaxe demandée.

¹ <https://nlp.stanford.edu/software/CRF-NER.html>

Erreurs de l'OCR

Nous appliquons du NER sur les données extraites de l'OCR. Or, nous avons déjà constaté que les résultats de celui-ci sont approximatifs. Nous insérerions donc des données potentiellement erronées dans la base de données si nous ne les corrigeons pas.

C'est pour cela que nous pouvons envisager d'appliquer le Fuzzy String Matching aux éléments trouvés par le NER. Cela réduirait grandement le nombre d'opérations car nous comparerions un nombre d'éléments beaucoup moins important. Nous pourrions appliquer une correction sur base d'un dictionnaire (base de données de e-sculape).

Nous pourrons cependant, encore une fois, pas offrir cette correction aux personnes ne faisant pas parties de l'écosystème de e-sculape.

14.3 NLP pour reconnaître le type de document

Etant donné qu'un humain aura tendance à classifier des documents en se basant sur des mots clés contenus dans celui-ci, une approche utilisant l'analyse de texte pour déduire la classe d'un document peut être envisageable.

Je m'intéresse dans ce projet, uniquement à une approche basée sur l'analyse d'images pour reconnaître le type de document. Il pourrait être envisageable pour la suite de tester ces méthodes et de les comparer aux résultats obtenus avec des CNN. Deux modèles "state of the art" peuvent s'avérer être intéressants :

- **BERT¹**
- **ELMo²**

¹ <https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>

² <https://allennlp.org/elmo>

15 Analyse d'image

15.1 Ressources

Je reste très abstrait dans ce chapitre. Pour en savoir plus sur le fonctionnement de l'algorithme CNN, des réseaux de neurones artificielles, du machine learning et du deep learning en général, je vous invite à aller lire les ressources suivantes qui m'ont grandement aidé pour comprendre ce domaine très vaste et complexe :

- Le site : <https://towardsdatascience.com/>
- Le livre : *Fundamentals of Deep Learning – Nikhil Buduma*
- Le livre : *Hands-On Machine Learning with Scikit-Learn and TensorFlow – Aurélien Géron*
- Le livre : *The Hundred-Page Machine Learning Book – Andriy Burkov's*
 - eBook : <http://thmlbook.com/wiki/doku.php>

15.2 Document classification

Nous avons vu comment différentes méthodes d'analyse de texte peuvent nous aider à trouver des métadonnées d'un document et à les classifier. Ces méthodes sont plus adaptées à l'extraction d'informations textuelles comme des noms ou des dates plutôt que pour trouver un nom de type de document.

En effet, le type de document peut ne pas figurer sur un des documents et donc sera absent de l'OCR (si par exemple le document comporte plusieurs pages et que seule la première comporte le type de document). Il nous faut donc un moyen de trouver le type de document qui soit indépendant de son contenu textuel. C'est pour cela qu'il est préférable de se tourner vers l'analyse d'image pour cette partie du processus, celle-ci analyse des pixels plutôt que du texte et reconnaît le type du document grâce à sa structure.

Cette approche se découpe totalement du NLP, elle utilise des modèles similaires à ceux utilisés pour la classification d'images. Les algorithmes de Deep Learning sont donc mis à l'honneur, nous allons voir comment nous pourront les utiliser dans notre projet.

(6) (7)

15.3 Image classification

15.3.1 Convolutional Neural Network (CNN)¹

Les modèles de classifications d'image les plus performants de nos jours utilisent tous des variantes de convolutional neural network (CNN), c'est un algorithme de Deep Learning. Il prend en input une image, y extrait des "features", assigne une importance à celles-ci en ajustant des poids et des biais et utilise ensuite ce qu'il a ajusté et appris pour faire des prédictions. Son fonctionnement est grandement inspiré par celui du cortex visuel humain et s'inspire donc d'un processus biologique.

Le Deep Learning, par rapport au Machine Learning, ajoute justement une phase automatique d'extraction de features à utiliser pour la classification rendant le système plus proche d'une approche humaine et bien sûr, plus indépendant. C'est nécessaire pour l'analyse de données très complexes comme des images car il peut être très difficile voire impossible pour un humain d'en extraire des features utilisables et pertinentes pour la prédiction.

Le Machine Learning reste encore largement utile pour des applications où il faut analyser des données comme des chiffres ou des textes où il est facile de les catégoriser afin d'entrainer un modèle. Il est beaucoup moins gourmand en ressources GPU.

(8) (9) (10)

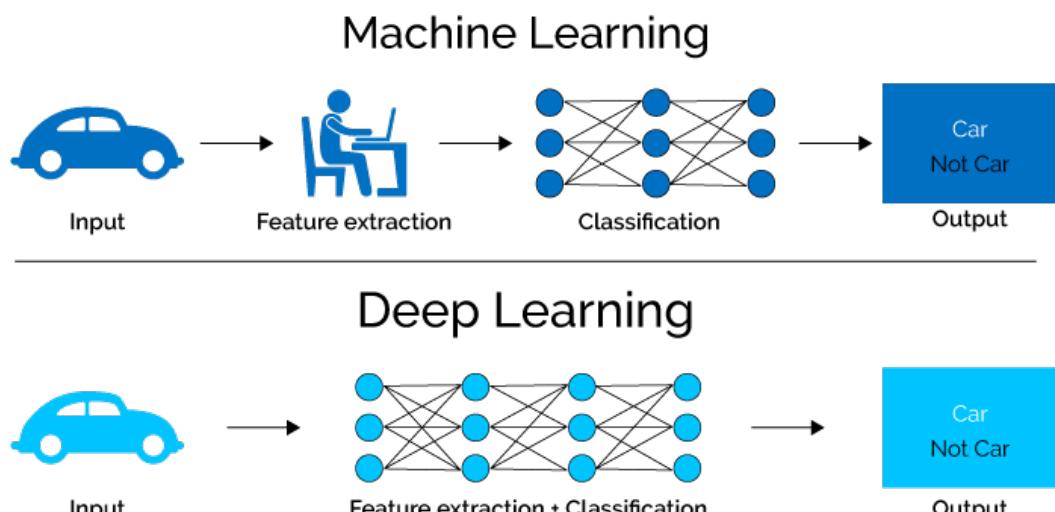


Figure 15.1 : Visualisation de la différence entre ML et DL²

¹ <https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb>

² https://cdn-images-1.medium.com/max/800/1*ZX05x1xYgaVoa4Vn2kKS9g.png

Fonctionnement succinct

Le but ultime de l'algorithme CNN est, comme nous l'avons vu, d'extraire automatiquement des features d'une image qu'il pourra ensuite utiliser pour faire de la reconnaissance. Il est capable d'être entrainé et d'apprendre en fonction des données qui lui sont données grâce à la "forward propagation" et la "back propagation".

Il le fait en empilant dans un réseau de neurones artificielles des couches de perceptrons¹. Ce réseau de neurones est composé d'une couche d'entrée, d'une couche de sortie et de couches cachées entre deux. Ces couches cachées sont généralement composées de : *Convolutional layers*, *ReLU layers*, *pooling layers*, et *fully connected layers* (Explications ici : <http://cs231n.github.io/convolutional-networks/>).

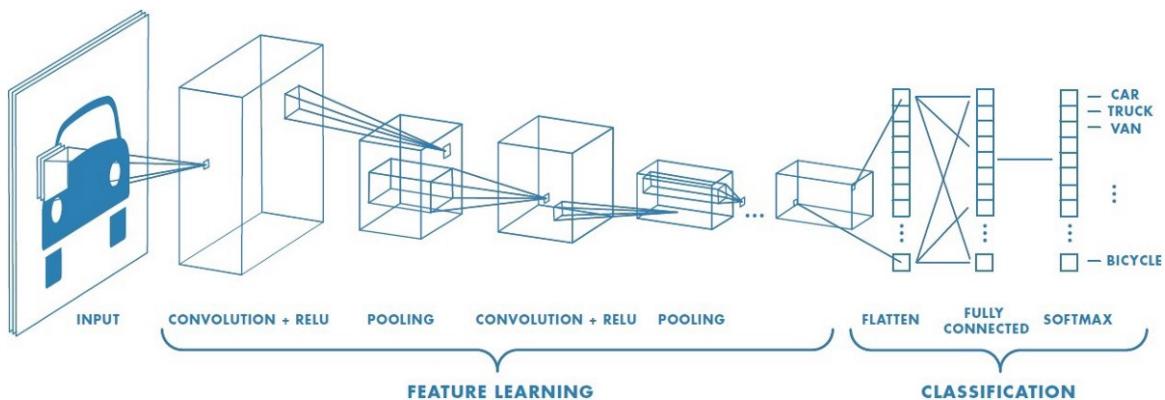


Figure 15.2 : Schématisation du fonctionnement d'un modèle CNN

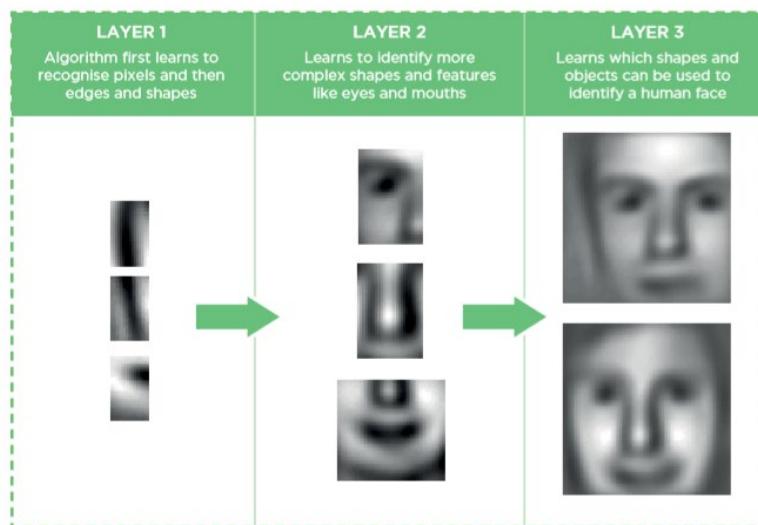


Figure 15.3 : Extractions de features par un algorithme CNN sur un visage humain²

¹ <https://fr.wikipedia.org/wiki/Perceptron>

² https://cdn-images-1.medium.com/max/800/1*KYUUg9JC6InYe-VNPMDzAA.png

15.3.2 Transfer learning¹

Il est très complexe et long de développer soi-même un CNN. Il est encore plus difficile de développer un CNN qui soit fiable et qui atteigne une efficacité de reconnaissance "state-of-the-art". Cela permet cependant de construire des modèles hautement spécifiques et permettant des résultats extrêmement fiables dans le domaine pour lequel ils ont été conçus.

C'est pour cela, à cause des contraintes de temps et mon manque d'expérience dans le domaine, que nous allons utiliser le "transfer learning" (ou "fine tuning"). Cela consiste à employer un CNN avec des fonctionnalités de reconnaissance d'images "général" existant et qui a fait ses preuves, de transférer ses connaissances et de le ré-entraîner sur un jeu de données personnalisé. Le but est qu'il ait un comportement spécifique à nos besoins, c'est-à-dire adapté à la reconnaissance de features de documents. Cette approche est un peu moins fiable qu'un système spécifique mais tous les exemples montrent que cela reste bien plus qu'acceptable et qu'on arrive à des précisions satisfaisantes.²

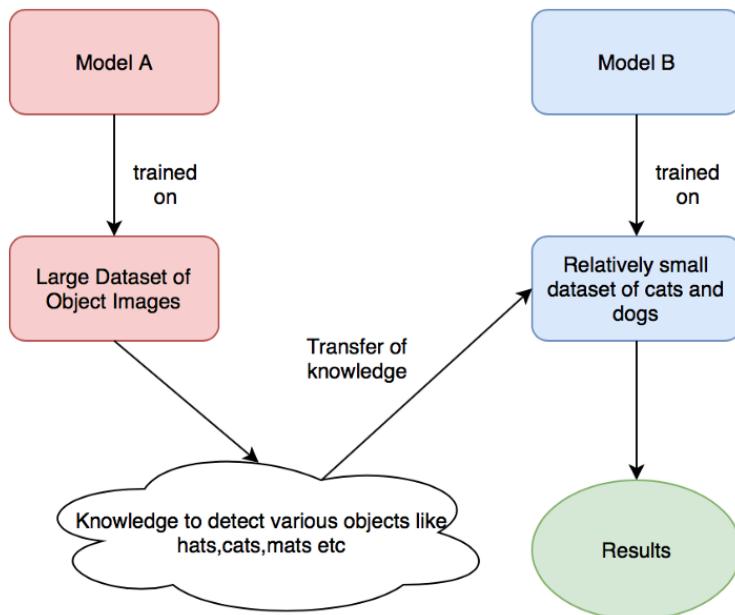


Figure 15.4 : Schématisation du transfer learning³

¹ <https://towardsdatascience.com/neural-network-architectures-156e5bad51ba>

² <https://www.youtube.com/watch?v=mPFq5KMxKVw>

³ <https://towardsdatascience.com/transfer-learning-using-differential-learning-rates-638455797f00>

Les architectures de CNN¹ envisageables sont les suivantes, elles sont toutes développées de base pour participer au challenge "ImageNet²" ou le but est d'entrainer un modèle sur un grand set d'images de choses de tous les jours et d'avoir la plus haute précision à les identifier sur des images de validation, je vais en lister quelques-unes qui ont gagné la compétition au fil des années et nous trouverons potentiellement une architecture idéale pour notre projet :

AlexNet

En 2012, premier modèle de DL utilisant un CNN participant au challenge. Il est réputé pour avoir été le premier à énormément augmenter le niveau de précision de ses prédictions grâce à cette architecture. Son succès fut le début d'une petite révolution dans le domaine du DL, tous les participants se sont mis à utiliser des architecture CNN les années suivantes.

VGG by Oxford

En 2014, c'est le premier modèle qui a obtenu un taux d'erreurs inférieur à 10%. Il sert de base à de nombreux autres modèles. ResNet et Inception reprennent certaines de ses idées.

Son problème est qu'il nécessite énormément de ressources, il est très lent.

ResNet by Microsoft

Encore une petite révolution., en 2015, ResNet introduit le "skip connection". Cela permet d'avoir des réseaux très profonds sans avoir d'énorme impacte sur les performances.

Inception-v4 by Google

Gagnant en 2016, invaincu à ce jour. C'est une évolution de l'ancienne architecture de Google (GoogleLeNet, Inception-v3) qui a été mergé avec l'idée proposée par ResNet. Il est légèrement plus gourmand en ressources que RestNet mais a une fiabilité sans égale.

¹ <https://towardsdatascience.com/neural-network-architectures-156e5bad51ba>

² <http://www.image-net.org/challenges/LSVRC/>

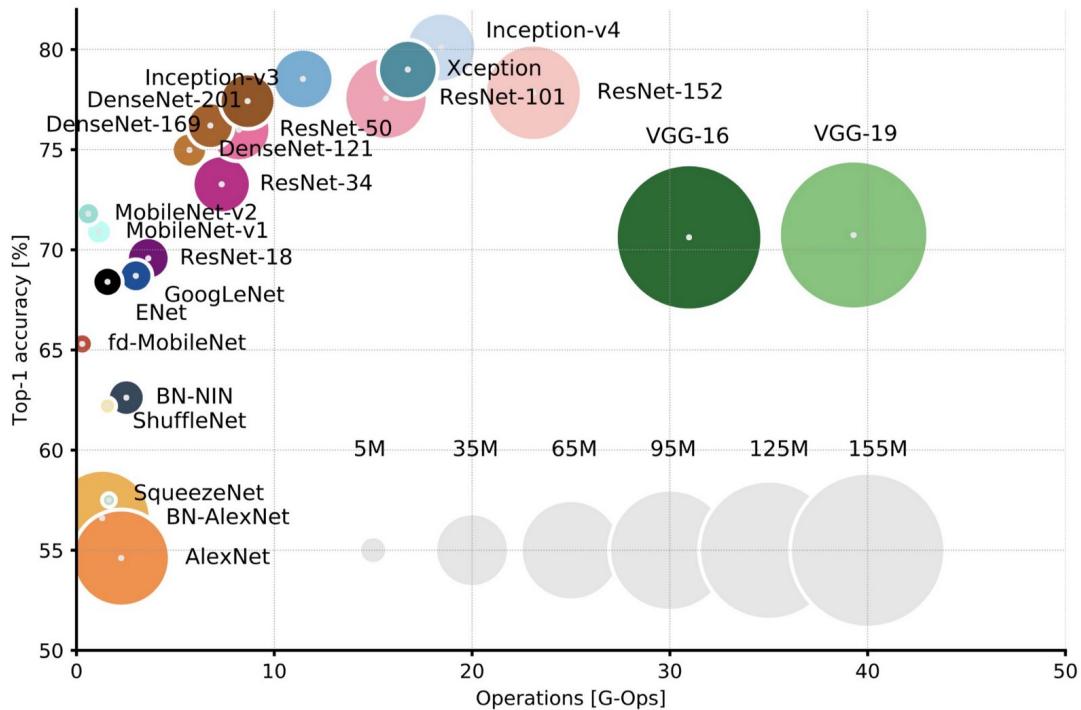


Figure 15.5 : Comparaison des différentes architectures CNN¹

Nous pouvons constater que deux architectures sortent leur épingle du jeu : **ResNet-101**, **Xception** et **Inception-V4**. Nous allons utiliser en réalisation la bibliothèque Keras qui n'implémente que certains de ces modèles comme Xception, Inception-V3, VGG-16, ResNet-50. VGG-16 a montré de bons résultats dans ce genre de champ d'applications dans de nombreuses études.

Est-ce que les modèles de CNN les plus performants le restent lors d'un transfer learning ?

15.3.3 Training Dataset

Pour ré-entrainer un modèle et utiliser le principe du "transfer learning", il nous faut des données d'entraînement et de test que nous pourrons donner à l'algorithme afin qu'il apprenne à prédire des données spécifiques que nous lui demandons plutôt que d'avoir un fonctionnement général. Nous avons vu dans le chapitre : Confidentialité ci-dessus les raisons pourquoi les données fournies par e-sculape ne sont pas adapté à une approche d'analyse d'image.

¹ https://cdn-images-1.medium.com/max/800/1*n16lj3lSkz2miMc_5cvkrA.jpeg

C'est pour pallier ces problèmes que nous utiliserons des dataset existants et dédiés à l'entraînement de classification de documents. Nous fournirons ainsi un modèle fonctionnel et testé afin de prouver que cette approche fonctionne. E-sculape pourront ensuite le ré-entraîner avec leurs propres documents et classes, un modèle de CNN par docteur. Cela élimine en plus le besoin qu'aurait eu e-sculape de nous fournir un grand nombre de données et permet de garantir la confidentialité des informations des patients.

RVL-CDIP Dataset¹

Dataset contenant 400'000 images de divers documents en noir/blanc divisés en 16 classes avec 25'000 images par classes. 320'000 images d'entraînement, 40'000 de validation et 40'000 de test. Sa taille excessivement grande pourrait poser des problèmes sur notre installation de taille modeste.

Les meilleurs scores obtenus sur ce dataset et les moyens utilisés pour y arriver se trouvent sur ce lien : <https://paperswithcode.com/sota/document-image-classification-rvl-cdip>

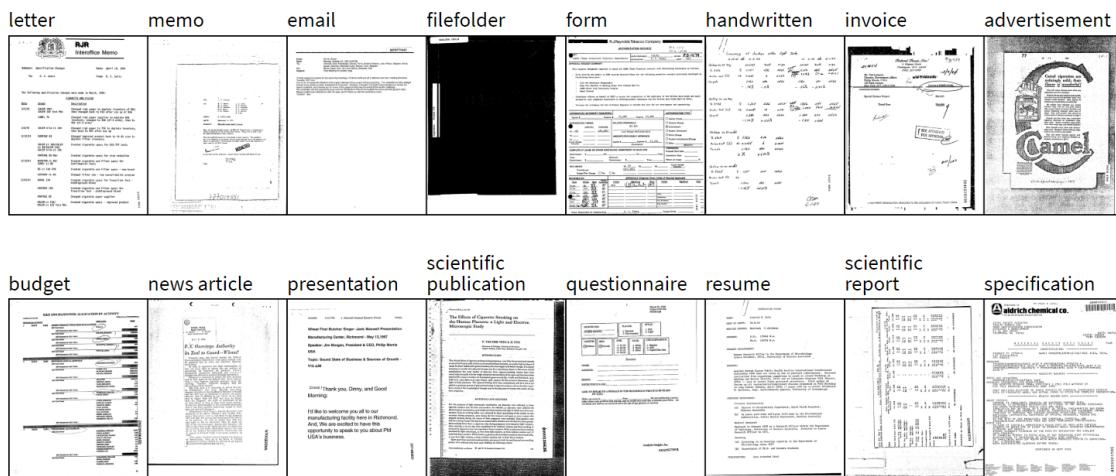


Figure 15.6 : Classes de documents et exemple pour chaque du RVL-CDIP Dataset

¹ <http://www.cs.cmu.edu/~aharley/rvl-cdip/>

Tobacco3482 Dataset¹ (11)

Dataset contenant 3'482 images de documents divisés en classes. Ces documents proviennent des dossiers publics des poursuites contre les compagnies de tabac américaines. Il possède des classes avec très peu d'exemplaires, à voir si cela suffit à obtenir une bonne fiabilité. On peut constater que le nombre de samples par classe varie beaucoup, il faudra évaluer si cela a une influence sur la fiabilité ou non.

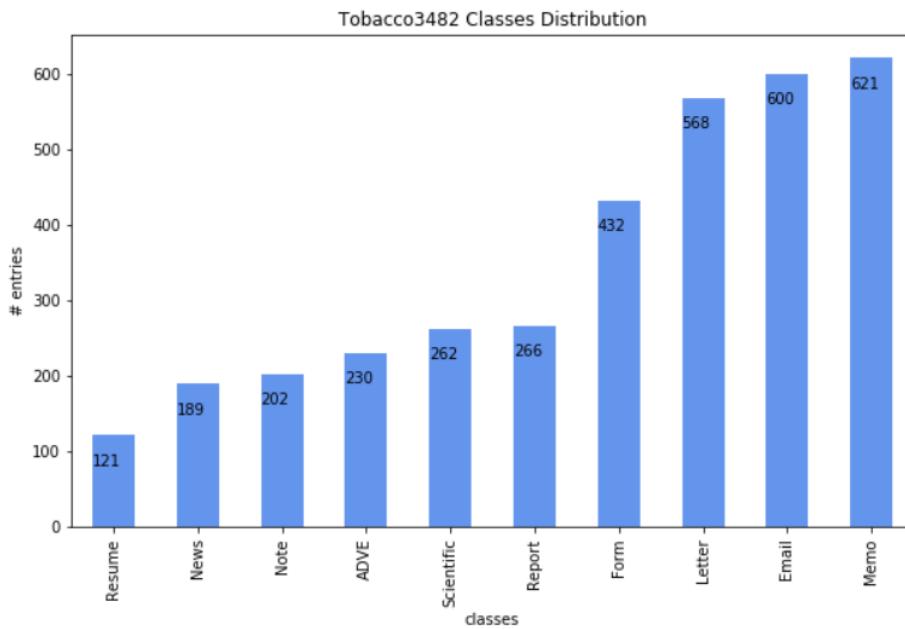


Figure 15.7 : Répartition des classes pour le dataset Tobacco3482

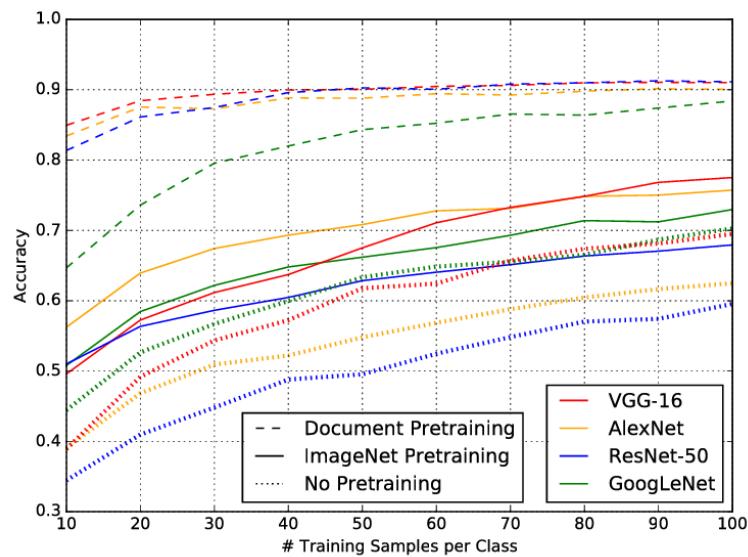


Figure 15.8 : Précision obtenue sur le dataset Tobacco3482 en fonction de diverses méthodes de transfer learning et différents modèles de CNN (12)

¹ <https://lampsrv02.umiacs.umd.edu/projdb/project.php?id=72>

15.3.4 Problèmes potentiels

Regrouper les documents

Si, par exemple, il y a plusieurs lettres dans un dossier médical. Que certaines de ces lettres font plusieurs pages et qu'elles ont toutes des dates différentes. Comment faire, vu que la numérisation sort un fichier PDF par document scanné, pour regrouper ensemble les lettres qui sont de la même date ? Comment ne pas les confondre ?

On pourrait, par exemple, utiliser les données extraites de l'OCR pour le faire. Regrouper les documents d'une classe possédant la même date par exemple.

CNN n'a pas une assez bonne fiabilité

Les potentielles causes qui rendraient un CNN inefficace seraient un nombre de documents pas assez grand pour permettre un bon entraînement où des documents beaucoup trop similaires visuellement et ne permettant pas au CNN d'en extraire des features permettant de les différencier.

Si le niveau de fiabilité n'est pas suffisant, il faudra se tourner vers une autre approche, par exemple, le NLP qui utiliserait le texte plutôt que l'image pour reconnaître la classe d'un document.

15.4 Triplet Network

15.4.1 One Shot Learning

Dans notre cas de figure, il se peut que nous n'ayons pas une grande quantité de données à disposition, du moins pas assez pour apprendre des features et entraîner convenablement un CNN. C'est dans ce genre de situation que nous pouvons nous tourner vers les méthodes dites de "One Shot Learning"¹. Ce sont des techniques permettant d'apprendre depuis un unique échantillon. Ils fonctionnent sur le principe suivant : construire une fonction de similarités qui compare deux images et indique s'il y a un "match".

¹ <https://towardsdatascience.com/siamese-network-triplet-loss-b4ca82c1aec8>

15.4.2 Triplet Loss

Un Triplet Network utilise la méthode du "Triplet Loss" (le "loss", ou perte correspond à une pénalité pour une mauvaise prédiction, plus le loss est petit, plus le système est fiable) pour être entraîné et fonctionne en comparant les similarités de 3 images. Une image d'ancrage, une image similaire (positive) et une image différente (négative). Les similarités entre l'image d'ancrage et l'image négative doivent être faibles et celles avec l'image positive doivent être grandes.

Cette méthode nous permet de calculer les gradients et de mettre à jour les poids et les biais du réseau. Le Triplet Loss minimise la "distance" entre l'ancrage et le positif d'une même entité et maximise la distance entre l'ancrage et la négative d'une autre entité.

(13)

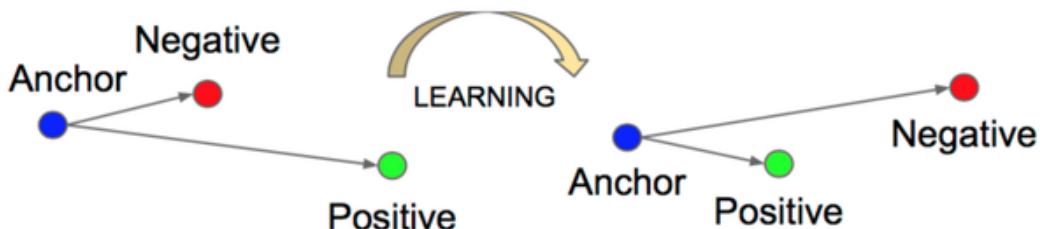


Figure 15.9 : Schématisation du fonctionnement de l'apprentissage via Triplet Loss¹

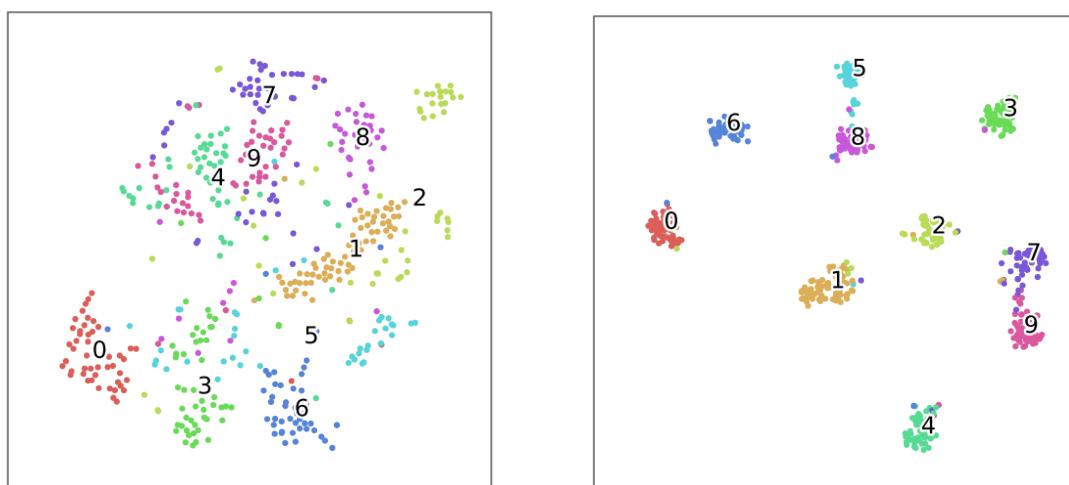


Figure 15.10 : Visualisation de la distribution des données avant (gauche) et après (droite) l'apprentissage d'un Triplet Network²

¹ https://cdn-images-1.medium.com/max/1600/0*_WNBFcRVEOz6QM7R.

² <https://github.com/KinWaiCheuk/Triplet-net-keras>

C'est une solution qui serait très intéressant de tester dans ce projet. En effet, notre cas est typiquement un cas où nous n'avons pas forcément beaucoup de données pour entraîner un CNN. C'est par contre, malheureusement, une méthode relativement compliquée à mettre en œuvre et relativement récente, il serait compliqué de l'implémenter dans le temps imparti.

15.5 MorphNet

Permet d'automatiser le design des structures de réseaux neuronaux. Permet d'optimiser les paramètres et de construire de NN plus rapides et plus petits. Au lieu de tester un grand nombre de possibilités d'architecture de réseau et de les comparer, il part d'un réseau existant (ResNet par exemple) résolvant un problème similaire et l'optimise pour la tâche à accomplir.

Son objectif est de diminuer la part de "chance" et de "trial and error" qui est aujourd'hui en vigueur lorsque l'on souhaite entraîner un modèle de deep learning qui soit fiable.¹

Cela pourrait être très intéressant dans notre projet. En effet, nous avons un modèle qui peut être trop grand, trop profond, trop lourd et non optimisé à cause du manque de temps pour tweaker les paramètres ainsi que de mon manque d'expérience. Il serait très intéressant de s'y pencher plus mais le temps imparti est malheureusement réduit.

(14)

Synthèse

Avec cet éventail de technologies, nous constatons qu'il n'est pas facile de faire un choix. Le critère premier étant la fiabilité, il faudra tester les différents modèles afin de tirer une conclusion.

Une approche **hybride** combinant analyse de texte (Fuzzy String Matching et NER) et analyse d'image (CNN Fine-tuning) semble être nécessaire pour mener à bien le projet.

¹ <https://towardsdatascience.com/with-morphnet-google-helps-you-build-faster-and-smaller-neural-networks-586e0baf7c36>

V. CONCEPTION

Introduction

Nous avons pu constater dans le chapitre précédent qu'il est nécessaire d'avoir une approche hybride pour espérer mener à bien le projet. Je vais, dans ce chapitre, modéliser une architecture imaginable dans le cadre d'une intégration dans le système existant de e-sculape et faire un éventail des librairies et des environnement utilisés pour le développement

16 Workflow et architecture

16.1 Phase d'entraînement

Avant de faire une prédiction, un CNN doit être entraîné. Il sera du devoir de e-sculape d'entraîner le modèle avec les données souhaitées. Les documents médicaux ayant une grande divergence entre cabinets, il est nécessaire d'entraîner **un modèle de CNN par médecin** ou cabinet afin de prédire ses documents.

Workflow :

1. Trier les PDF scannés dans différents dossiers labélisés en fonction des classes
2. Les convertir de PDF (output du scannage) en JPG
3. Donner le dataset ainsi créé au CNN afin de l'entraîner
4. Evaluer les performances
5. Sauver le modèle entraîné pour l'utiliser lors de la prédiction

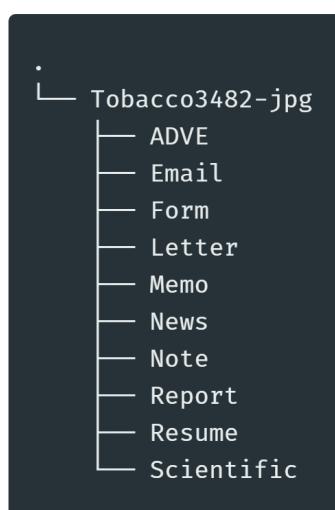


Figure 16.1 : Structure du dataset avec les dossiers labelisés contenant les images

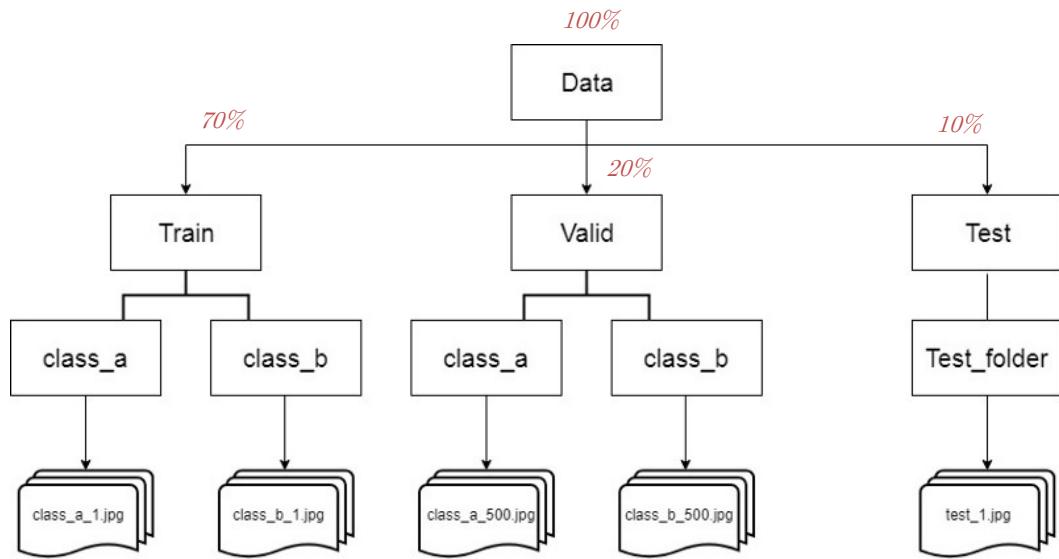


Figure 16.2 : Structure de fichiers utilisé par le CNN pour l'entraînement (fait de manière autonome par l'algorithme)¹

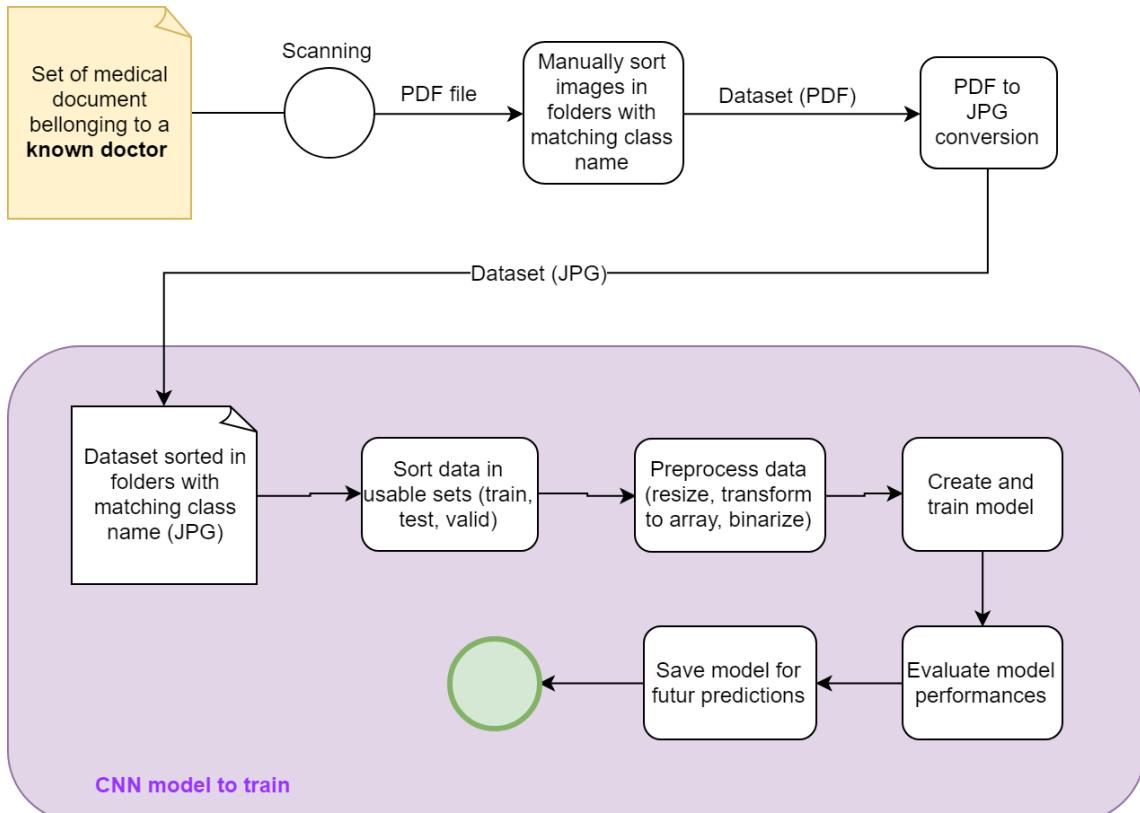


Figure 16.3 : Schéma du workflow d'entraînement du modèle

¹ https://cdn-images-1.medium.com/max/800/1*Hpvpa9pBJXKxaPCL5tKnLg.jpeg

16.2 Phase de prédition

L'approche hybride nous fait créer plusieurs "modules" qui interagissent entre eux et s'envoient des données le long d'un cycle de vie afin de prédire le type de document et d'en extraire toutes les informations intéressantes. Les voici :

1. Extraction de l'OCR en texte

Permet de fournir des données utilisables en format ".txt" pour le NER et le Fuzzy String Matching

2. Transformation PDF en JPG

Transforme les fichiers scannés au format PDF en JPG. Cela nous donne des données que notre CNN pourra reconnaître.

3. Pretrained CNN model

Utilise un modèle préalablement pré-entraîné afin de reconnaître la classe des documents qui lui sont donnés.

4. Named Entity Recognition

Permet d'extraire de manière grossière les informations pertinentes de l'OCR.

5. Regrouper les documents

Regroupe-les documents de la même classe ayant plusieurs pages entre eux.

6. Fuzzy String Matching

Permet de corriger les erreurs de l'OCR et de retrouver les informations importantes en faisant des "match" avec la base de données.

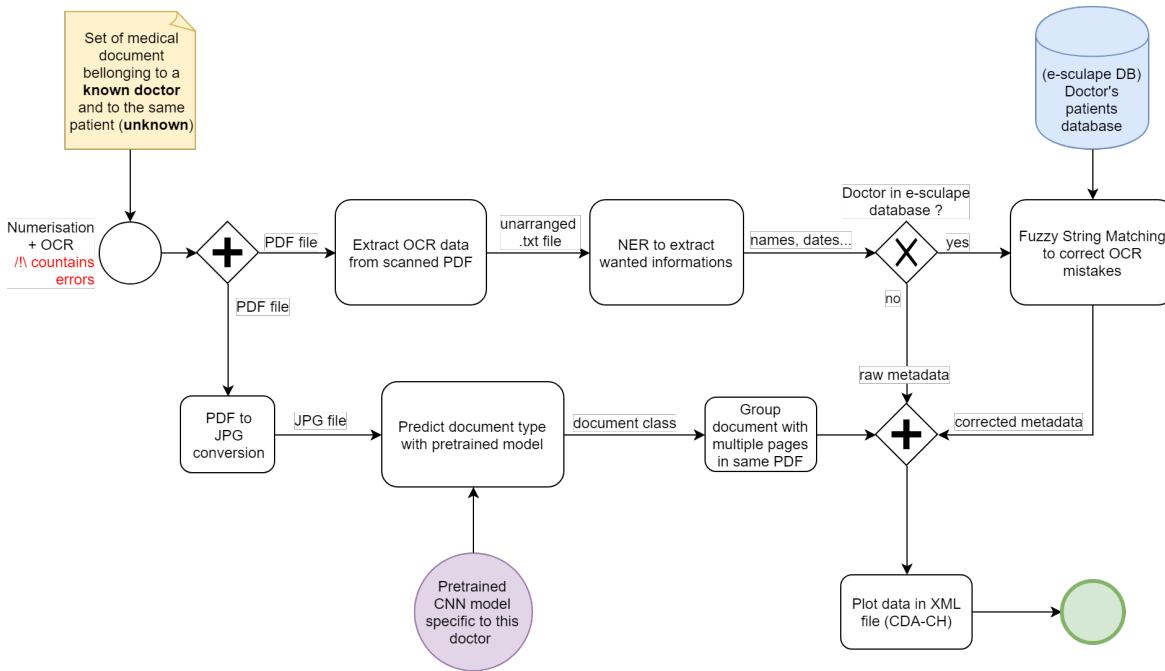


Figure 16.4 : Schéma de la nouvelle architecture qui va être mise en place

17 Environnement

17.1 Kaggle¹

C'est une plateforme web et une communauté consacrée à la data science, au machine learning et au deep learning. Elle met à disposition de ses utilisateurs un grand nombre de datasets sur lesquels les utilisateurs peuvent s'entraîner, y effectuer des challenges et avoir recours à la communauté pour les aider. On peut également y uploader des dataset et consulter les différentes solutions proposées par la communauté pour s'en inspirer.

J'ai choisi cette plateforme car elle utilise des Notebook Jupyter² comme environnement de développement. C'est un "coteau suisse" du développement Python qui contient toutes les bibliothèques couramment utilisées pour faire du machine learning (pandas, cv2, PIL, Keras, Tensorflow, numpy, matplotlib...). Il nous permet de créer des documents interactifs composés de blocs de code et de texte formaté.

¹ <https://www.kaggle.com/>

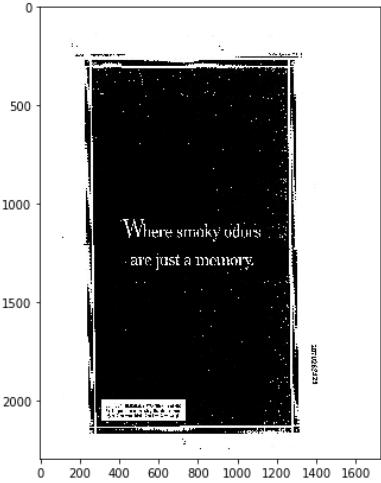
² <https://jupyter.org/>

Elle met également gratuitement à disposition des GPU Nvidia P100 pour entraîner son modèle ce qui est essentiel dans l'analyse d'image si l'on souhaite avoir des temps de calcul qui restent raisonnables. Cela permet également de ne pas avoir besoin d'une grosse workstation pour développer et de tout faire via un ordinateur portable connecté à internet.

Il est bien sûr possible d'exporter le travail effectué sur cet environnement de développement afin de l'utiliser ou l'on souhaite. L'export est soit un fichier .ipynb ou un script Python.

Plot data

```
In[140]:  
random.Random(seed).shuffle(total_set)  
  
for ima in total_set[0:3] :  
    print(ima)  
    img = mpimg.imread(ima)  
    plt.figure(figsize=(7,7))  
    imgplot = plt.imshow(img, cmap="gray")  
    plt.show()  
  
..../input/tobacco3482-jpg/Tobacco3482-jpg/ADVE/2070262428_24  
29.jpg
```



Sessions

Draft Session | 6h:52m/9h | GPU On ●
CPU 0.06%
RAM 5.9GB/13GB
Disk 323.5MB/4.9GB

Workspace

input (read-only) X
Tobacco3482 X

Versions

Settings

Figure 17.1 : Exemple de Notebook Jupyter montrant du texte formaté, un bloc de code, la console avec le résultat et une figure

18 Deep Learning Library

18.1 Keras¹

Keras est une librairie Python haut-niveau créée par M. François Chollet, ingénieur chez Google, ayant pour but de rendre le développement de modèles de deep learning plus intuitif et plus orienté vers l'humain (plus d'abstraction). Keras est, pour ces raisons, la meilleure porte d'entrée dans le domaine du deep learning pour une personne novice. Dans sa version 2.0, Tensorflow intègre nativement Keras.

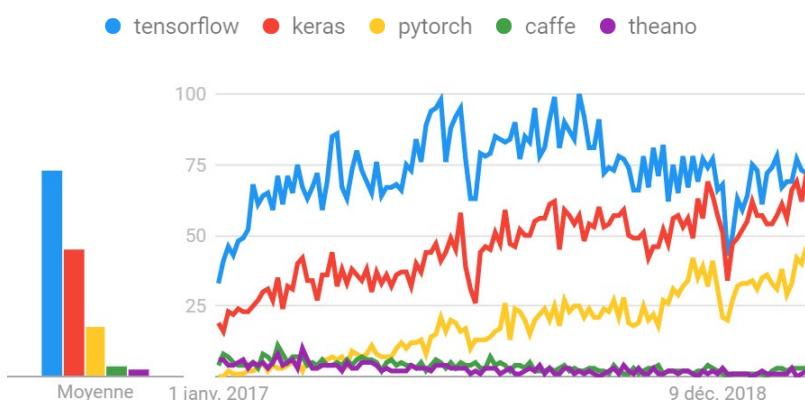


Figure 18.1 : Evolution de l'intérêt pour différents framework

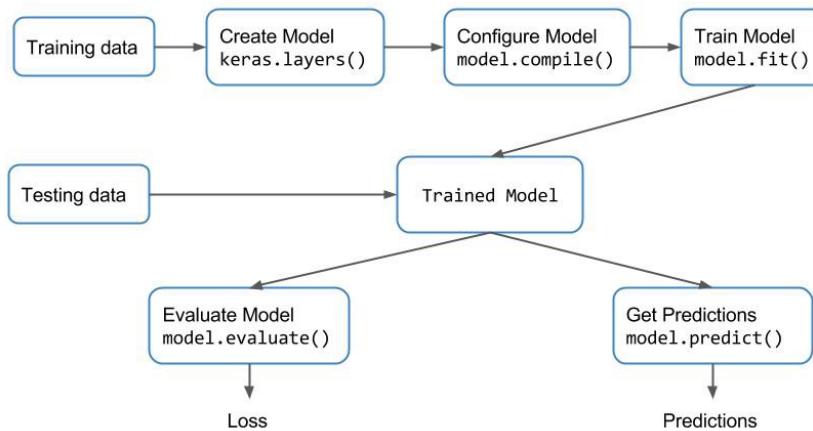


Figure 18.2 : Keras Workflow² afin de créer, entraîner et utiliser un modèle

¹ <https://keras.io/>

² <https://www.learnopencv.com/wp-content/uploads/2017/09/keras-workflow.jpg>

VI. RÉALISATION

Introduction

Je vais développer, dans ce chapitre, les différents "modules" qui composeront l'architecture de la solution étudiée en conception. Les modules restent pour le moment indépendants les uns des autres. Il faudra à terme les faire interagir entre eux et les intégrer dans le système existant de e-sculape.

19 Extraction de l'OCR en texte

Software utilisé : **pdftotext**¹

Convertie tous les PDF contenu dans le dossier "input" (et ses sous-dossiers) en fichiers TXT et les déplace dans le dossier "output" (y compris dans les bons fichiers).

```
mkdir output;
cp -r input/* output/;
for file in output/**/*.pdf; do pdftotext "$file" "$file.txt"; done;
for file in output/**/*.pdf; do rm "$file"; done;
```

Figure 19.1 : Script bash utilisant pdftotext pour faire la batch conversion



Figure 19.2 : Exemple de document PDF scanné

¹ <https://linux.die.net/man/1/pdftotext>

```
.FR

nôpital fribourgeois
freiburger spital

HFR Riaz
Rue de l'Hôpital 9
1632 Riaz
Secrétariat médical
T. 026 919 9116
F. 026 919 9117

Service de chirurgie

Docteur
Test MEDECIN
Rte. d'Englisberg 13
1763 Granges-Paccot

Médecin chef
Médecins adjoints

Riaz, le 18.05.2010 JM/dv

Monsieur et cher Confrère,
Votre patient, Monsieur Patient MONDEMO, né le 22.01.1956, a séjourné dans notre service
de chirurgie du 09.05.2010 au 13.05.2010, date de son retour à domicile.
```

Figure 19.3 : Résultat obtenu sur le même document après extraction de l'OCR en format texte

20 Transformation PDF en JPG

Software utilisé : **convert** de **imagemagick**¹

Convertie tous les PDF contenu dans le dossier "input" (et ses sous-dossiers) en fichiers JPG et les déplace dans le dossier "output" (y compris dans les bons fichiers).

```
mkdir output;
cp -r input/* output/;
for file in output/*/*.pdf; do convert -density 300 "$file" -quality 100 "$file.jpg"; done;
for file in output/*/*.pdf; do rm "$file"; done;
```

Figure 20.1 : Script bash utilisant convert pour faire la batch conversion

¹ <https://imagemagick.org/script/convert.php>

21 Transfer Learning from CNN¹

Voir annexe : 2. Kaggle Jupyter Notebook

Référence :

- Le site : <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>
- Le livre : *Deep Learning With Python – François Chollet* (15)

Etapes

1. Trier les données dans des sets
2. Transformer les données
3. Créer le modèle de CNN
4. Ajouter au modèle les couches personnalisées
5. Evaluer le modèle
6. Sauver le modèle

21.1 Trier les données dans des sets

Comme vu dans le chapitre : Phase d'entraînement ci-dessus, les données du dataset doivent être organisés en une structure précise afin d'être utilisable pour l'entraînement. Les sets suivants doivent être créées :

- **Train (70% du dataset)** : utilisé durant la phase d'entraînement pour entraîner le modèle et ajuster les poids et les biais. Il contient des images labellisées.
- **Validation (20% du dataset)** : Pour évaluer le modèle durant la phase d'entraînement. Utilisé pour "fine-tuner" les hyperparamètres du modèle. Le modèle n'apprend donc pas à partir de ces données. Il contient des images labellisées.
- **Test (10% du dataset)** : Utilisé pour donner une évaluation non-biaisée des performances en termes de fiabilité du modèle. Contient des images sans label.

¹ <https://keras.io/applications/#applications>

Nous allons utiliser, pour la réalisation, le dataset Tobacco3482 précédemment documenté. Il contient au total **3'482** images que nous séparons en : **2'437** images d'entraînement, **696** images de validation et **342** images de test. Il est important de noter que nous sommes dans un cas où nous avons spécialement peu de données pour un entraînement de CNN mais cela représente la réalité de la mise en œuvre potentielle chez e-sculape.

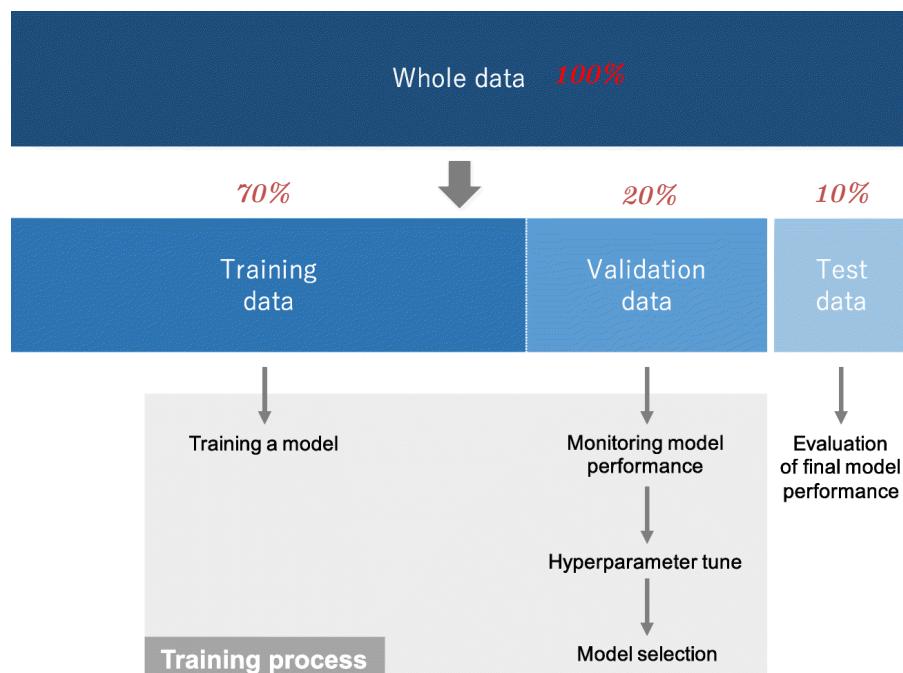
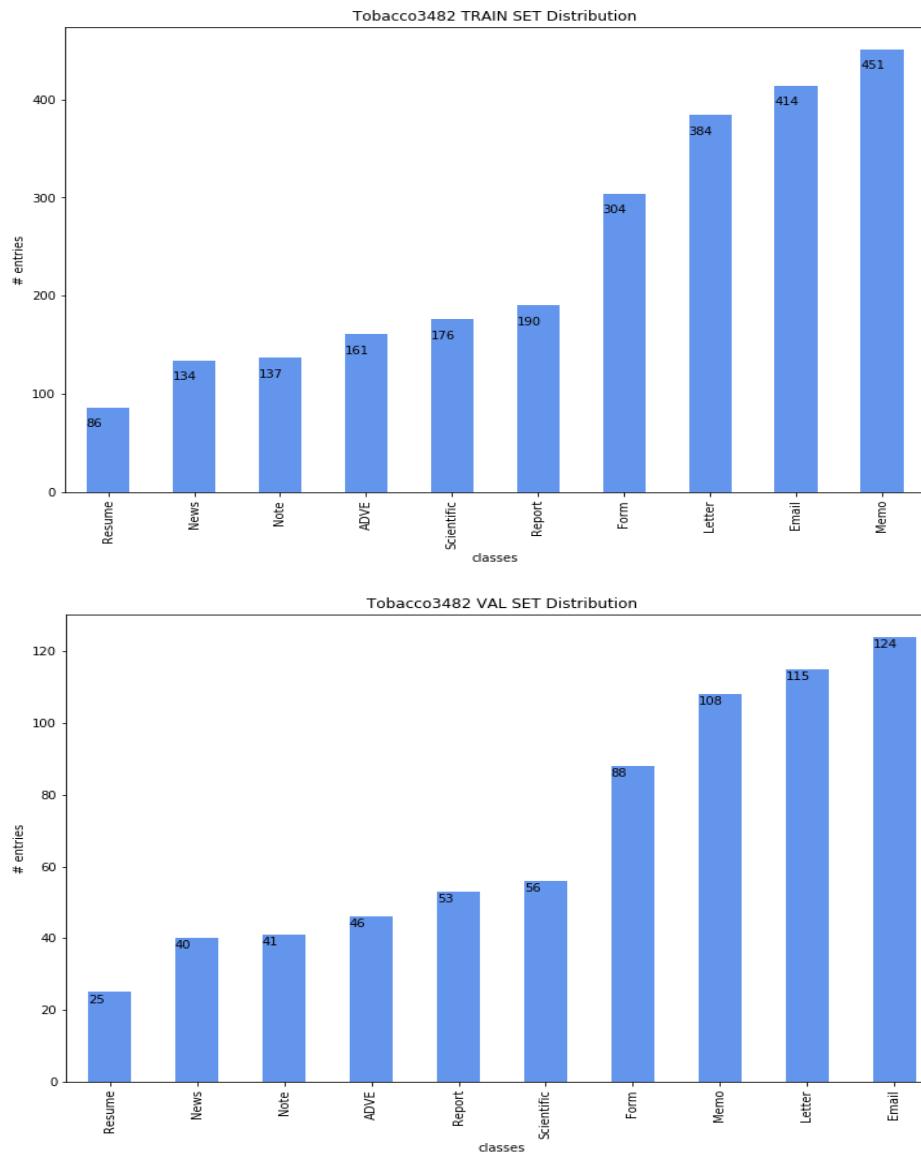


Figure 21.1 : Schématisation de la division du dataset et différents sous-sets¹

On peut voir ci-dessous, comme cité lors de la présentation du dataset qu'il y a une **grande variance du nombre de samples par classes**. J'effectuerai dans le chapitre

¹ <https://link.springer.com/article/10.1007/s13244-018-0639-9>

Tests et évaluation ci-dessous un test avec le même dataset étant "balancé" plus tard afin de montrer si cela a une influence sur la fiabilité ou non.



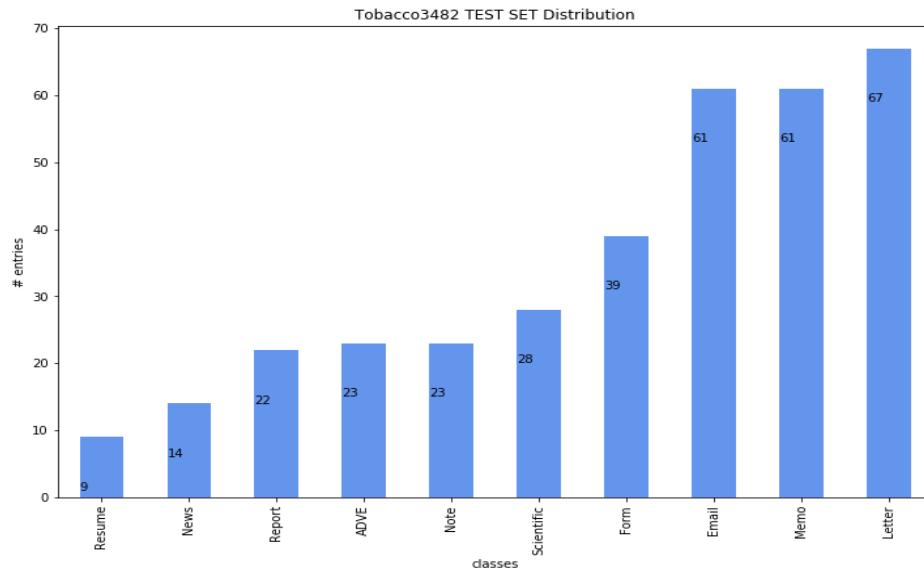


Figure 21.2 : Distribution des données dans les différents sets

21.2 Transformer les données

Nous utilisons Keras ainsi que des modèles de CNN existant, nous devons formater nos données d'une certaine manière afin de respecter leur syntaxe et les donner en entrée afin d'entrainer le modèle. Nous avons déjà toutes nos images séparées dans des sets sous formes de listes Python ainsi que les labels qui les accompagnent. Il faut à présent :

Redimensionner les images en fonction du modèle de CNN

Chaque modèle existant de CNN a des dimensions d'entrée définis. La forme de l'entrée est définie comme cela : (`img_width, img_height, channels`). Il est bien sur possible de modifier ces dimensions pour diminuer le temps de calcul (moins de pixels = moins de paramètres) mais durant mes testes, cela a uniquement tendance à diminuer la fiabilité du modèle et les gains en temps ne sont pas substantiels.

Le "channel" correspond aux nombres de canaux de couleurs, un pour une image en noir et blanc, trois pour une image en couleur (RGB). Dans notre cas, le channel sera toujours égal à 3 car les images scannées peuvent être en couleur.

Dimensions par défaut de l'input¹ :

CNN Model	Image Width (px)	Image Height (px)	Channels
VGG-16	224	224	3
ResNet-50	224	224	3

¹ <https://keras.io/applications/>

InceptionV3	299	299	3
Xception	299	299	3

```
def process_images(img_set) :
    processed_img = []

    for i in range(len(img_set)) :
        processed_img.append(cv2.resize(cv2.imread(img_set[i], cv2.IMREAD_COLOR),
                                         (img_size, img_size)))

    return processed_img
```

Figure 21.3 : Fonction utilisant opencv (cv2) pour redimensionner les images

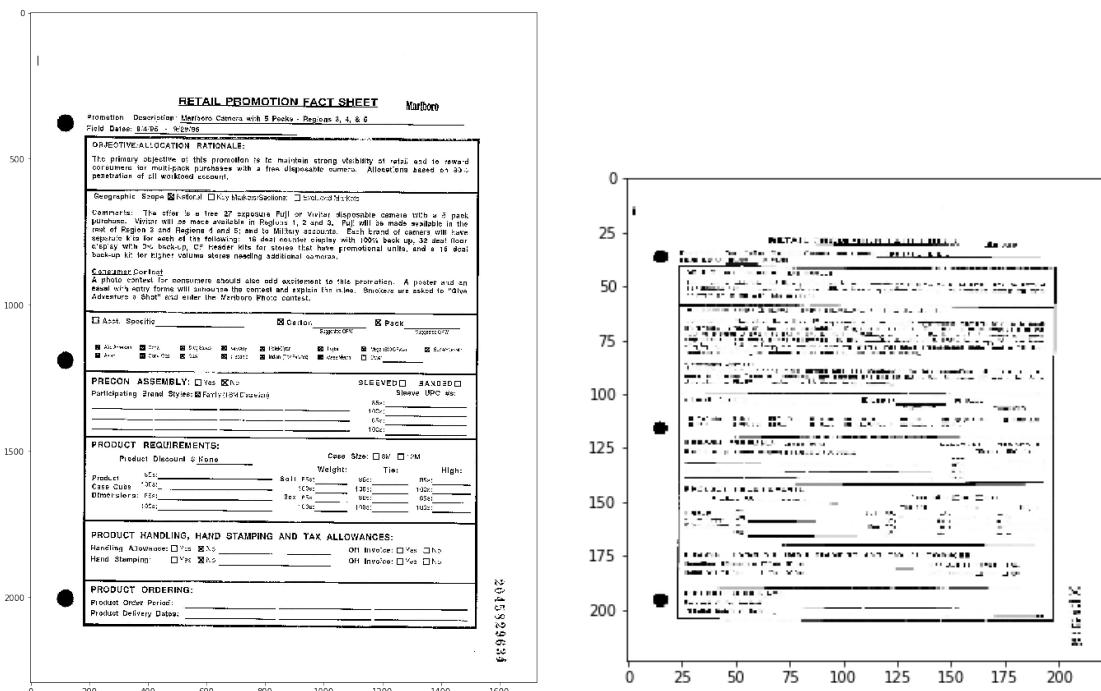


Figure 21.4 : Document avant et après redimensionnement

Transformer les listes Python en Numpy Array

La liste contentant les images redimensionnées ainsi que la liste contenant les labels doivent maintenant être transformées en Numpy Array pour être acceptés en entrée pour l'entraînement du modèle. Cela nous donne un tableau de `img_width*img_height*channels` contenant dans chaque case une valeur de 0 à 255 correspondants à l'intensité du pixel de couleur.

```
x_train = np.array(data_train)
y_train = lb.transform(np.array(train_label))

print("train images shape : ", x_train.shape)
print("train labels shape : ", y_train.shape)
print(x_train[0])
```

```
train images shape : (2437, 224, 224, 3)
train labels shape : (2437, 10)
[[[255 255 255]
 [255 255 255]
 [255 255 255]
 ...
 ...
```

Figure 21.5 : transforme la liste d'images et de labels en Numpy Array, print leur "forme" ainsi que la valeur des pixels

Le premier nombre de la forme (ou shape) correspond aux nombres d'éléments contenus dans l'array. On peut constater que les informations suivantes correspondent aux paramètres définis dans l'étape d'avant. Le "10" de `y_train` correspond aux nombres de classes. Les valeurs contenues dans la première case de `x_train` sont bel et bien des valeurs entre 0 et 255 (255 = blanc).

On peut également constater que la liste des labels subit une transformation supplémentaire (`lb.transform`), je l'explique ci-dessous :

Binariser les sets de labels avec le one hot encoding¹

Les labels, qui sont sous forme de texte, doivent être transformés en représentation numérique pour être compris par le modèle. On crée un binarizer contenant toutes les classes et on l'utilise sur notre liste de labels. Cela nous donne, pour chaque label, une liste avec un 1 et des 0. L'ordre est le même que dans la liste des classes, l'index où se situe le 1 est le même que l'index où se situe le nom de la classe correspondante.

¹ [https://fr.wikipedia.org/wiki/Encodage one-hot](https://fr.wikipedia.org/wiki/Encodage_one-hot)

```
lb = LabelBinarizer()
lb.fit(list(classes))

y_train = lb.transform(np.array(train_label))

print(train_label[0])
print(y_train[0])
print(lb.classes_)
```

```
ADVE
[1 0 0 0 0 0 0 0 0 0]
['ADVE' 'Email' 'Form' 'Letter' 'Memo' 'News' 'Note' 'Report' 'Resume'
 'Scientific']
```

Figure 21.6 : Binarisation des labels : print avant et après transformation et print des classes contenues dans le binarizer

21.3 Créer le modèle de CNN¹

Etapes :

0. Définir les paramètres (description ci-dessous)
1. Créer le modèle de base depuis un CNN existant
2. Les couches de ce modèle de base sont entrainables
3. Créer le modèle customisé
4. Ajouter au modèle custom le modèle de base
5. Ajouter les couches en fonction des besoins (Flatten, Dropout, Dense...)
6. Compiler le modèle ainsi créé
7. Entrainer le modèle

Etant donné que la création des modèles est très similaire (à part les types couches ajoutées), je vais surtout m'attarder ici sur VGG-16 qui a montré les meilleurs résultats. Pour avoir de plus amples informations sur les autres modèles, leurs spécificités, ainsi que leurs couches, rendez-vous sur :

- **Informations sur les modèles de CNN** : <https://keras.io/applications/>
- **Informations sur les couches** : <https://keras.io/layers/about-keras-layers/>

La définition des paramètres et des couches ajoutées s'est faite de manière arbitraire en testant et en reprenant ce qui marchait bien (méthodologie "trial and error") et en s'inspirant de divers exemples de Kernels² sur Kaggle. C'est un travail de plusieurs jours pour fine-tuner un CNN afin qu'il fasse ce qu'on attende de lui et encore plusieurs autres jours supplémentaires afin qu'il produise des résultats d'une grande fiabilité. Un petit changement peut faire de grandes différences. Il existe cependant certaines indications sur la valeur des paramètres lors du transfer learning :

- Garder un learning-rate bas et augmenter le nombre d'epochs.
- Avoir un "batch size" assez grand et en puissance de 2.
- Utiliser une couche de dropout pour éviter l'overfitting.
- Utiliser la fonction d'activation³ "relu" pour les couches denses et "softmax" pour la couche de prédiction.

¹ <https://www.kaggle.com/bugraokcu/cnn-with-keras>

² <https://www.kaggle.com/kernels>

³ <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>

21.3.1 VGG-16¹

```

# 0) parameters
img_size = 224
batch_size = 32
epochs = 50
channels = 3
learning_rate = 0.00001

# 1) base model
base_model = VGG16(weights = "imagenet", include_top=False, input_shape = (img_size,
img_size, channels))

# 2) layers trainables
#for layer in base_model.layers:
#    layer.trainable = False

# 3) custom model
model = models.Sequential()

# 4) add baase model
model.add(base_model)
# 5) add layers
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu', name='dense'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(len(classes), activation='softmax', name='predictions'))

# 6) compile model
model.compile(optimizer=optimizers.Adam(lr=learning_rate), loss='categorical_crossentropy',
metrics=['accuracy'])

# 7) train model
train_model = model.fit(x_train, y_train,
                        batch_size=batch_size,
                        epochs=epochs,
                        verbose=1,
                        validation_data=(x_val, y_val))

```

Figure 21.7 : Création, compilation en entraînement d'un modèle VGG-16

0) Définition des paramètres :

- **Batch Size** : Nombre d'échantillons par mise à jour de gradient.
- **Epochs** : Un passage sur l'ensemble des données.
- **Learning Rate** : A quelle vitesse un modèle apprend un problème.

¹ <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>

1) Création du modèle de base :

- **VGG16** : on définit l'architecture de base comme étant VGG16
- **Weights** : on définit les poids du modèle pré-entraîné comme étant ceux entraînés sur "imagenet".
- **Include_top** : inclure ou non la couche entièrement connectée au sommet du réseau.
- **Input_shape** : définit la forme de l'entrée, doit matcher avec ce qui a été fait en 21.2 ci-dessus.

5) Ajout des nouvelles couches :

- **Flatten** : aplatis l'entrée.
- **Dense** : couche du NN entièrement connectée.
- **Dropout** : consiste à définir de manière aléatoire un taux de désactivation des neurones pour éviter l'overfitting.
- **Activation** : fonction d'activation, "Relu" pour la couche Dense, "Softmax" pour la couche de prédiction. Défini si un neurone va s'activer ou non.

6) Compiler le modèle

- **Optimizer** : Fonction pour produire des résultats légèrement meilleurs et plus rapides en mettant à jour les paramètres du modèle tels que les valeurs de poids et de biais.
- **Loss** : Fonction de minimisation du nombre d'erreurs
- **Metric** : Fonction utilisées pour juger la performance du modèle.

7) Entrainer le modèle

On lui donne en entrée les données d'entraînement, les labels d'entraînement, le batch size, le nombre d'epochs, une verbosité (1 = barre de progression) et des données de validation. Les données de validation servent à évaluer le modèle à la fin de chaque epoch.

Les phases de compilation de d'entraînement des différents modèles CNN s'effectue de la même manière.

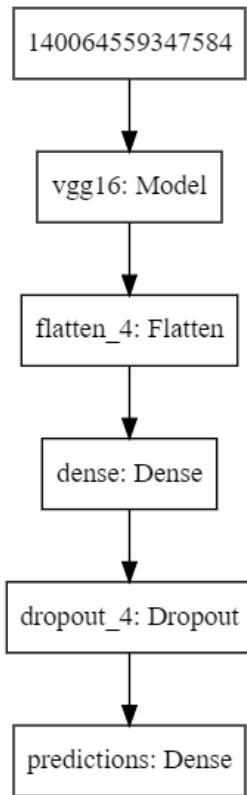


Figure 21.8 : Structure du modèle créé, on y retrouve toutes les couches que nous avons défini

21.3.2 ResNet-50

```

#base model
base_model = ResNet50(weights = "imagenet", include_top=False, input_shape = (img_size,
img_size, channels))

# custom model
model = models.Sequential()

# add layers
model.add(base_model)
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu', name='dense'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(len(classes), activation='softmax', name='predictions'))
  
```

Figure 21.9 : Crédit d'un modèle ResNet50

21.3.3 InceptionV3

```
#base model
base_model = InceptionV3(weights = "imagenet", include_top=False, input_shape = (img_size,
img_size, channels))

# custom model
x = base_model.output

# add layers
x = GlobalAveragePooling2D()(x)
x = Dense(128,activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(len(classes), activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)
```

Figure 21.10 : Création d'un modèle InceptionV3

21.3.4 Xception

```
#base model
base_model = Xception(weights = "imagenet", include_top=False, input_shape = (img_size,
img_size, channels))

# custom model
x = base_model.output

# add layers
x = GlobalAveragePooling2D()(x)
x = Dense(128,activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(len(classes), activation='softmax')(x)

# add your top layer block to your base model
model = Model(base_model.input, predictions)
```

Figure 21.11 : Création d'un modèle Xception

21.4 Evaluer le modèle

Une méthode intégrée à Keras nous permet de mesurer les performances du modèle une fois entraîné. Il mesure les performances sur le "test set" :

```
score = model.evaluate(x_test, y_test, verbose=1)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

Figure 21.12 : Test de la fiabilité (accuracy) et des erreurs (loss)

```
Test loss: 0.9885052019993235
Test accuracy: 0.8472622471515314
```

- **Accuracy** : 84.7%
- **Loss** : 0.98

On peut également visualiser l'évolution de la fiabilité et des erreurs au fil des epochs. La ligne bleue représente le set d'entraînement et la ligne orange le set de validation. Le set de validation n'a pas une autant bonne fiabilité ($\sim 85\%$) que le set d'entraînement ($\sim 99\%$) car ce dernier est hautement biaisé à cause de sa faible généralisation. C'est la fiabilité du set de validation qui est intéressant pour cette raison. Plus les deux lignes sont éloignées, plus le modèle overfit. On peut également constater que ~ 50 epochs auraient suffis à avoir un modèle avec la même fiabilité et aurait réduit de 2 fois le temps d'entraînement.

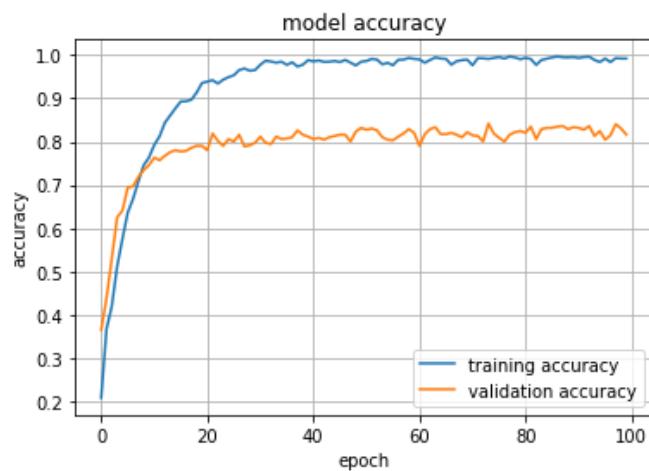
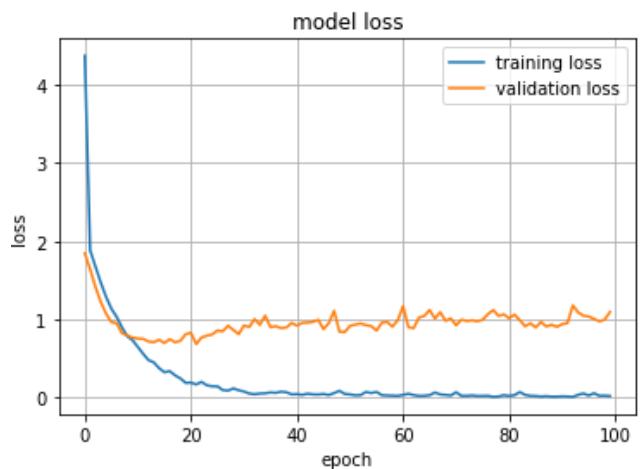


Figure 21.13 : Evolution de l'accuracy et du loss au fil des epochs

21.5 Sauver le modèle

Pour réutiliser le modèle pour faire des prédictions sur d'autre dataset, il faut le sauver et l'exporter. Le modèle sauvé contient : l'architecture du modèle permettant de le recréer, les poids et la configuration (loss, optimizer, paramètres).

```
model.save('trained_model.h5')
```

Figure 21.14 : Sauver le modèle

22 Named Entity Recognition

Il a finalement été décidé d'utiliser SpaCy, il propose le support du français de manière native, est très bien documenté et est relativement facile à mettre en place et à ré-entrainer.

Informations à extraire des documents "anonymisés" de e-sculape :

- **Nom du docteur** : Test Medecin
- **Nom du patient** : Monsieur Patient Mondemo
- **Date de naissance du patient** : 22.01.1956
- **Hôpital** : HFR Riaz

22.1 Modèle existant

Il est assez facile d'extraire des informations de l'OCR au moyen de SpaCy. Nous prenons chaque fichiers texte de chacun des dossier (dans "input"), y appliquons le NER et les sauvegardons au format JSON dans un dossier "output".

22.1.1 Français

```

input = 'input/'
output = 'output/fr/'

nlp = spacy.load("fr")

data = {}
data['MISC'] = []
data['PER'] = []
data['LOC'] = []
data['ORG'] = []

for path, dir, files in os.walk(input):
    for file in files:
        if file.endswith(".txt"):
            filename = (os.path.join(path, file))
            ocr = open(filename).read()
            doc = nlp(ocr)
            for ent in doc.ents:
                data[ent.label_].append(ent.text)
            if not os.path.exists(output+path):
                os.makedirs(output+path)
            with open(output+filename+'.json', 'w') as json_file:
                json.dump(data, json_file)

```

Figure 22.1 : Script Python utilisant SpaCy pour faire du NER en batch sur l'OCR en batch et sauvegarder le résultat en JSON

HFR Riaz

Rue de l'Hôpital 9

1632 Riaz MISC Secrétariat médical

T. LOC 026 919 9116

F. LOC 026 919 9117

Service de chirurgie ORG Docteur MISC Test MEDECIN

Rte MISC . d' Englisberg LOC 13

1763 Granges-Paccot PER Médecin chef

Médecins LOC adjoints

Riaz LOC , le 18.05.2010 JM/dv

Monsieur et cher Confrère LOC ,

Votre LOC patient, Monsieur Patient MONDEMO PER , né le 22.01.1956, a séjourné dans notre service de chirurgie du 09.05.2010 au 13.05.2010, date de son retour à domicile.

Figure 22.2 : Résultat du NER français sur le texte de l'OCR¹

¹ <https://spacy.io/usage/linguistic-features#named-entities-101>

22.1.2 Anglais

```
{
  "ORG": [
    "Confrère",
    "Monsieur Patient MONDEMO",
    "du 09.05.2010",
    "Le 10.05.2010",
    "Allergie",
    "Status",
    "Status",
    "BAV du Ier"
  ],
  "PERSON": [
    "Rue de l'Hôpital",
    "Rte",
    "Granges-Paccot",
    "Hyperplasie",
    "Cataracte de l'œil",
    "Anémie"
  ],
  "DATE": [
    "9",
    ",",
    "1763",
    "13.05.2010",
    "1993",
    "2007"
  ],
}
```

Figure 22.3 : Output sous format JSON avec la classe comme clé et les valeurs correspondantes

HFR Riaz

Rue de l'Hôpital PERSON 9 DATE 1632 Riaz

Secrétariat médical

T. 026 919 9116

F. 026 919 9117

Docteur

Test MEDECIN

Rte PERSON . d'Englisberg 13 CARDINAL

1763 DATE Granges-Paccot PERSON

Médecin chef

Médecins adjoints

Riaz, le 18.05.2010 JM/dv

Monsieur et cher Confrère ORG ,

Votre patient, Monsieur Patient MONDEMO ORG , né le 22.01.1956, a séjourné dans notre service de chirurgie du 09.05.2010 ORG au 13.05.2010 DATE , date de son retour à domicile.

Figure 22.4 : Visualisation du résultat du NER en anglais sur le texte de l'OCR

Les résultats des modèles de base laissent vraiment à désirer. Nous pouvons constater dans les exemples ci-dessus que les modèles existants, tant français que anglais, effectuent beaucoup d'erreurs. Aucun ne parvient à extraire et catégoriser convenablement les informations que nous souhaitons.

C'est pourquoi je vais essayer d'utiliser un modèle entraîné pour la reconnaissance des informations spécifiques à nos documents médicaux et comparer les résultats.

22.2 Modèle ré-entraîné¹

Il faut créer un dataset comportant, par exemple, une liste des noms des médecins, des patients, des hôpitaux correctement tagués que nous pourrions ensuite utiliser pour entraîner notre système NER.

Un script permettant de ré-entraîner le modèle a été réalisé mais le temps imparti ne permet pas de construire un dataset adéquat. C'est pourquoi j'ai choisi d'en prendre un en ligne et de l'utiliser comme base. Un système personnalisé pour e-sculape pourra bien sûr reprendre la même forme.

Le script et le dataset proviennent de la source suivante :

<https://github.com/DataTurks-Engg/Entity-Recognition-In-Resumes-SpaCy>

C'est un projet visant à extraire grâce au NER les informations intéressantes d'un CV. Le dataset contient 220 CV annotés.

Etapes :

1. Créer un dataset grâce à l'outil d'annotation *Dataturks*² sur des textes de l'OCR
2. Télécharger le fichier JSON généré contenant les informations annotées
3. L'utiliser afin d'entraîner le système de NER fourni par SpaCY
4. Présenter au modèle des données non-annotées afin d'en faire des prédictions

¹ <https://spacy.io/usage/training>

² <https://dataturks.com/features/document-ner-annotation.php>

```

    "label": [
        "Skills"
    ],
    "points": [
        {
            "start": 743,
            "end": 1140,
            "text": "Database (Less than 1 year), HTML (Less than 1 year), Linux. (Less than 1 year), MICROSOFT\\nACCESS (Less than 1 year), MICROSOFT WINDOWS (Less than 1 year)\\n\\nADDITIONAL INFORMATION\\n\\nTECHNICAL SKILLS:\\n\\n"
        }
    ]
},
{
    "label": [
        "Graduation Year"
    ],
    "points": [
        {
            "start": 729,
            "end": 732,
            "text": "2016"
        }
    ]
},
{
    "label": [
        "College Name"
    ],
    "points": [
        {
            "start": 675,
            "end": 702,
            "text": "Shivaji University Kolhapur "
        }
    ]
}
]

```

Figure 22.5 : Exemple de fichier JSON contenant les données texte et leur classe attribuée (*label*)

Le but n'est bien sûr pas simplement de retrouver des mots dans un dictionnaire et de les matcher à une classe, nous voulons avoir un modèle **général** qui est capable de prédire la classe d'un mot en fonction de son contexte. C'est pour cela que les techniques de ML sont appliquées.

```

> python3 train.py
> python3 -m spacy train en models output_train_data.json output_dev_data.json

```

Figure 22.6 : Commandes à exécuter dans le terminal afin d'entrainer le modèle

Malheureusement, après de nombreux essaies et recherches, il m'est impossible d'effectuer l'entraînement. Une erreur nous est retournée et je ne suis pas parvenu à la résoudre dans le temps imparti.

Je ne suis donc pas parvenu à extraire les informations nécessaires au moyen de SpaCy. Il sera du devoir d'un étudiant reprenant ce projet de se pencher sur la question et de parvenir à entraîner le nouveau modèle de NER afin de le faire.

```

Traceback (most recent call last):
  File "/home/patrick/.local/lib/python3.6/site-packages/spacy/cli/train.py", line 253, in train
    scorer = nlp_loaded.evaluate(dev_docs, debug)
  File "/home/patrick/.local/lib/python3.6/site-packages/spacy/language.py", line 600, in evaluate
    docs, golds = zip(*docs_golds)
ValueError: not enough values to unpack (expected 2, got 0)

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/lib/python3.6/runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
  File "/usr/lib/python3.6/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "/home/patrick/.local/lib/python3.6/site-packages/spacy/_main_.py", line 38, in <module>
    plac.call(commands[command], sys.argv[1:])
  File "/home/patrick/.local/lib/python3.6/site-packages/plac_core.py", line 328, in call
    cmd, result = parser.consume(arglist)
  File "/home/patrick/.local/lib/python3.6/site-packages/plac_core.py", line 207, in consume
    return cmd, self.func(*args + varargs + extraopts), **kwargs)
  File "/home/patrick/.local/lib/python3.6/site-packages/spacy/cli/train.py", line 301, in train
    best_model_path = _collate_best_model(meta, output_path, nlp.pipe_names)
  File "/home/patrick/.local/lib/python3.6/site-packages/spacy/cli/train.py", line 344, in _collate_best_model
    bests[component] = _find_best(output_path, component)
  File "/home/patrick/.local/lib/python3.6/site-packages/spacy/cli/train.py", line 361, in _find_best
    accs = srsly.read_json(epoch_model / "accuracy.json")
  File "/home/patrick/.local/lib/python3.6/site-packages/srsly/_json_api.py", line 49, in read_json    file_path = force_path(location)
  File "/home/patrick/.local/lib/python3.6/site-packages/srsly/util.py", line 11, in force_path
    raise ValueError("Can't read file: {}".format(location))
ValueError: Can't read file: models/model0/accuracy.json

```

Figure 22.7 : Erreur retournée

23 Fuzzy String Matching

Le but est de retrouver les informations du texte de l'OCR grâce à un matching fait avec les informations contenues dans une base de données. Deux méthodes précédemment étudiées sont utilisées et comparées : Levenshtein et Jaro.

Le fonctionnement est le suivant :

1. On récupère l'OCR d'un fichier scanné
2. On récupère les informations de la base de données des médecins et des patients
3. On itère à travers tous les mots de ceux-ci
4. On mesure entre chacun la distance afin de trouver des "match"

```
[  
  {  
    "N° Enr.": 1,  
    "IDPatInt": "TEST11",  
    "PatNom": "MONDEMO",  
    "PatPrenom": "Patient",  
    "PatDDN": "19560122",  
    "PatAdresse": "rte. d'Englisberg 13",  
    "PatNPA": 1763,  
    "PatLieu": "GRANGES-PACCOT",  
    "PatTel": 265500580,  
    "PatSex": "M"  
  },  
  {  
    "N° Enr.": 2,  
    "IDPatInt": "TEST12",  
    "PatNom": "PONSE",  
    "PatPrenom": "Ray",  
    "PatDDN": "19560122",  
    "PatAdresse": "rte. d'Englisberg 14",  
    "PatNPA": 1763,  
    "PatLieu": "GRANGES-PACCOT",  
    "PatTel": 265500581,  
    "PatSex": "M"  
  },  
  {  
    "N° Enr.": 1,  
    "IDMedecin": 1,  
    "MedTitre": ["Dr. "],  
    "MedNom": ["AUDRIAZ"],  
    "MedPrenom": ["Patrick"],  
    "MedEAN": [760100022111],  
    "MedRCC": ["A123456"],  
    "MedAdresse": ["rte. d'Englisberg 13"],  
    "MedNPA": [1763],  
    "MedLieu": ["Granges-Paccot"],  
    "MedTel": ["026 550 05 80"],  
    "MedFax": ["026 550 05 81"],  
    "MedCabinet": ["Cabinet du Dr. Audriaz Patrick"]  
  },  
  {  
    "N° Enr.": 2,  
    "IDMedecin": 2,  
    "MedTitre": ["Dr. "],  
    "MedNom": ["MEDECIN"],  
    "MedPrenom": ["Test"],  
    "MedEAN": [760100022112],  
    "MedRCC": ["A123457"],  
    "MedAdresse": ["rte. d'Englisberg 14"],  
    "MedNPA": [1763],  
    "MedLieu": ["Granges-Paccot"],  
    "MedTel": ["026 550 05 80"],  
    "MedFax": ["026 550 05 81"],  
    "MedCabinet": ["Cabinet du Dr. Test Medecin"]  
  }]
```

Figure 23.1 : Exemple d'entrées dans la DB du patient et du médecin au format JSON contenant les entrées que nous pourrons matcher avec les mots de l'OCR

23.1 Levenshtein

Utilisation de la librairie Python : **fuzzywuzzy**¹

On affiche que les éléments ayant obtenu un score supérieur à 80% afin de trier les résultats et retourner seulement ceux qui sont cohérents.

```
for e in stringToFind:  
    # Fuzzy string matching with Levenshtein Distance  
    if (fuzz.token_sort_ratio(word.lower(), e) > 80):  
        print("search : ", e)  
        print("found : " + word)  
        print("score : ", fuzz.token_sort_ratio(word.lower(), e))  
        print("file : ", filename)
```

Figure 23.2 : Utilisation de la librairie pour trouver les matchs au moyen de la distance de Levenshtein

¹ <https://github.com/seatgeek/fuzzywuzzy>

```
> python find.py

('search : ', 'medecin')
found : MEDECIN
('score : ', 100)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : Médecin
('score : ', 92)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : Médecins
('score : ', 86)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : MEDECIN
('score : ', 100)
('file : ', 'input/bbb/Classeur1 5.pdf.txt')
-----
```

Figure 23.3 : Résultat obtenu en recherchant le nom des médecins de la DB dans l'OCR des fichiers données en entrée avec affichage de la distance (score)

23.2 Jaro

Utilisation de la librairie Python : **textdistance**¹

On affiche que les éléments ayant obtenu un score supérieur à 0.8 afin de trier les résultats et retourner seulement ceux qui sont cohérents.

```
for e in stringToFind:
    # Fuzzy string matching with Jaro Distance
    if (textdistance.jaro.normalized_similarity(word.lower(), e) > 0.8):
        print("search : ", e)
        print("found : " + word)
        print("score : ", textdistance.jaro.normalized_similarity(word.lower(), e))
        print("file : ", filename)
```

Figure 23.4 : Utilisation de la librairie pour trouver les matchs au moyen de la distance de Jaro

¹ <https://pypi.org/project/textdistance/>

```
> python find.py

('search : ', 'medecin')
found : MEDECIN
('score : ', 1.0)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : Médecin
('score : ', 0.8492063492063493)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : Médecins
('score : ', 0.8134920634920636)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : MEDECIN
('score : ', 1.0)
('file : ', 'input/bbb/Classeur1 5.pdf.txt')
-----
```

Figure 23.5 : Résultat obtenu en recherchant le nom des médecins de la DB dans l'OCR des fichiers données en entrée avec affichage de la distance (score)

Synthèse

Nous avons pu voir dans ce chapitre un tour d'horizon des différents "modules" qui composeront l'architecture. Un peu de travail reste à faire afin de les faire fonctionner en harmonie entre eux et non pas de manière indépendante. Il y a également le système de NER qui ne peut pas être utilisé tel quel et qui doit être ré-entraîné afin d'afficher une fiabilité convenable.

VII. TESTS ET ÉVALUATION

24 Performances des CNN

Les performances des différents modèles de CNN développés ont été évaluées sur le dataset Tobacco3482 en utilisant la plateforme Kaggle et le framework de ML Keras.

24.1 Fiabilité

CNN Model	Top Accuracy	Loss	Différence
VGG-16	84.7 %	0.98	+ 10.1 %
ResNet-50	80.9 %	0.94	+ 5.2 %
InceptionV3	80.4 %	0.81	+ 4.5 %
Xception	76.9 %	0.96	+ 0 %

Nous pouvons constater que **VGG-16** est 10.1% plus fiable que Xception, la fiabilité étant cruciale dans notre cas d'application, nous le choisirons donc. ResNet-50 et InceptionV3 sont également acceptables mais pas à un niveau "state-of-the-art".

24.1.1 Visualisation de la fiabilité

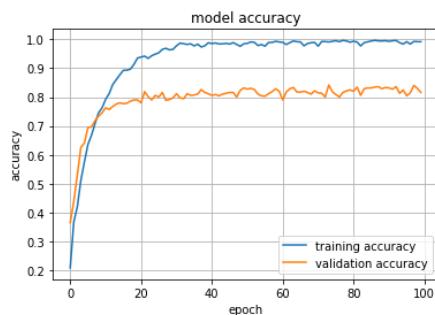
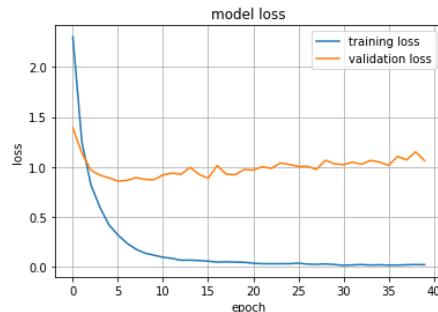
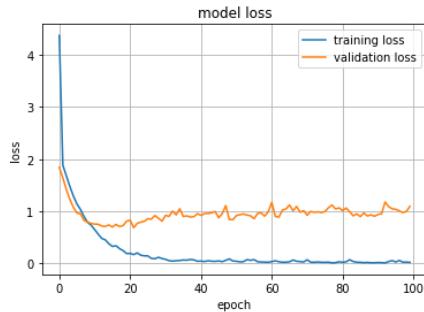


Figure 24.1 : VGG-16



Figure 24.2 : ResNet-50

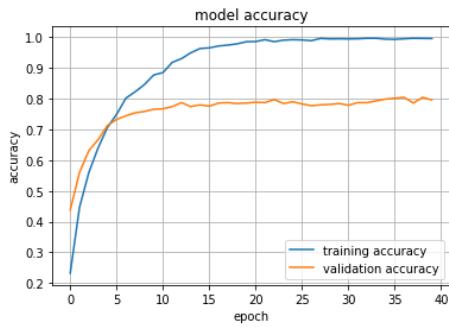
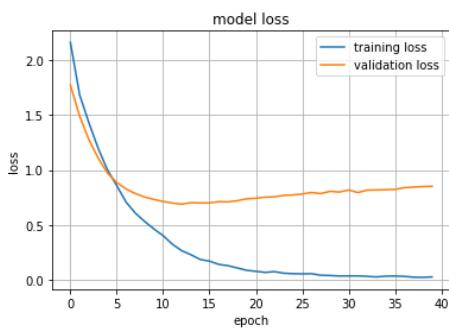


Figure 24.3 : InceptionV3

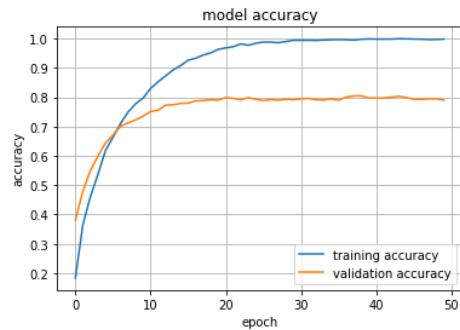
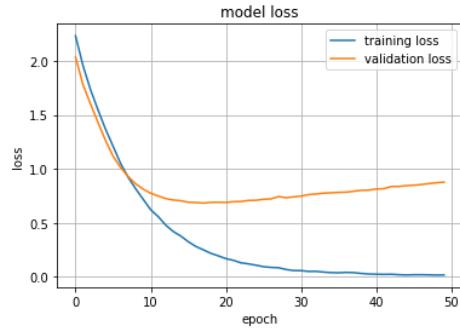


Figure 24.4 : Xception

24.2 Temps d'entraînement total

CNN Model	# trainable weights	# epochs	Time/epoch	Total Training Time
ResNet-50	216	40	19 sec	760 sec (12.6 min)
InceptionV3	192	40	27 sec	1'080 sec (18 min)
VGG-16	30	100	19 sec	1'900 sec (31.6 min)
Xception	158	50	65 sec	3'250 sec (54.2 min)

Les gains en temps d'entraînement total ne sont pas des métriques qui ne sont très importantes par rapport à la fiabilité. Le temps d'entraînement total n'influence donc pas beaucoup le choix du modèle. ResNet-50 reste cependant le second en termes de fiabilité et n'est donc pas à exclure.

Nous pouvons, grâce à ces diverses constatations, choisir **VGG-16** comme modèle le plus approprié à notre cas d'utilisation. **ResNet-50** peut être intéressant si le temps vient à manquer pour l'entraînement ou que là nous ne disposons pas d'une machine très puissante.

24.3 Améliorer la fiabilité du CNN

24.3.1 Evaluation selon la forme des courbes

En observant la forme des courbes de fiabilité, on peut déduire des informations qui sont précieuses et donnant des indications sur le tweaking des hyperparamètres permettant de mener à un modèle plus fiable. J'ai utilisé cette méthode dans un stade initial afin d'avoir un modèle qui soit cohérent.

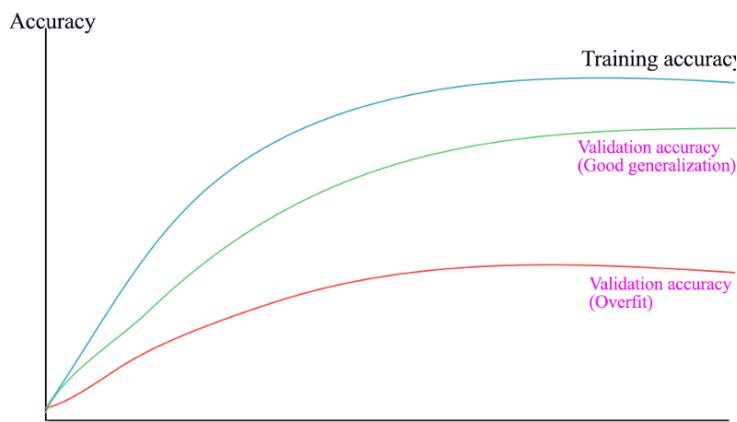


Figure 24.5 : Schématisation de courbes qui indiquent un overfitting du modèle, notre modèle n'overfit pas¹

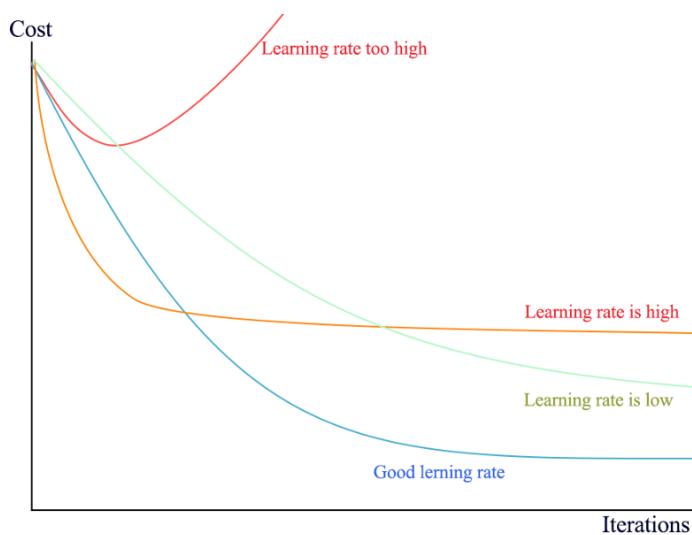


Figure 24.6 : Evaluation du learning rate en fonction de la forme de la courbe, le nôtre est adéquat.²

¹ https://medium.com/@jonathan_hui/improve-deep-learning-models-performance-network-tuning-part-6-29bf90df6d2d

² https://medium.com/@jonathan_hui/debug-a-deep-learning-network-part-5-1123c20f960d

24.3.2 Evolution de la fiabilité sur CNN-16



Figure 24.7 : "Deep learning is an art, not a science"¹

Il m'a fallu de nombreuses heures de test, de tweakage, des paramètres, de documentation², de trial & error afin de, petit à petite, obtenir un modèle le plus fiable possible avec 84.7 % de fiabilité.

Premier test :

- Fiabilité : **66.7%**
- Base model trainable : non
- Paramètres entraînables : 525,002
- Temps d'entraînement total : 6.6 min
- Taille de l'image : 150 x 150
- Train size / validation size / test size : 0.6/0.2/0.2
- Batch size : 16
- Learning rate : 0.001
- Epochs : 100
- Dropout : 0.2
- Dernière couche Dense : 64

¹ <https://xkcd.com/1838/>

² <https://blog.floydhub.com/guide-to-hyperparameters-search-for-deep-learning-models/>

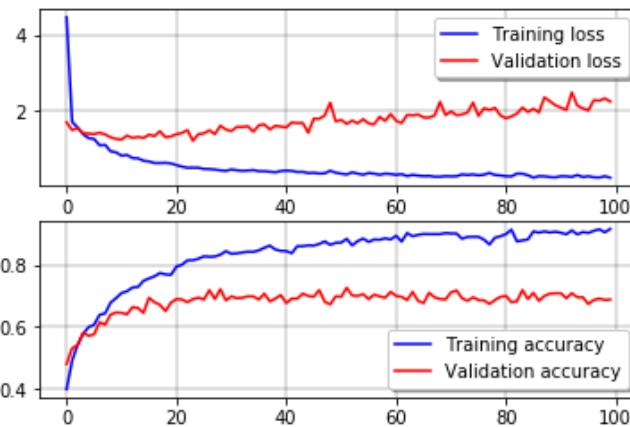


Figure 24.8 : Graphique d'évolution de la fiabilité et des erreurs au fil du temps

Second test :

- Fiabilité : **77.5%**
- Base model trainable : non
- Paramètres entraînables : **3,212,682**
- Temps d'entraînement total : **13.3 min**
- Taille de l'image : **224 x 224**
- Train size / validation size / test size : **0.7/0.2/0.1**
- Batch size : **32**
- Learning rate : **0.00001**
- Epochs : 100
- Dropout : **0.5**
- Dernière couche Dense : **128**

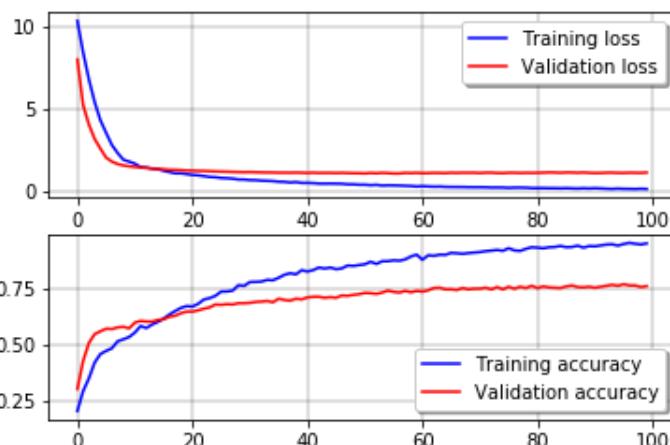


Figure 24.9 : Graphique d'évolution de la fiabilité et des erreurs au fil du temps

Troisième test (final) :

- Fiabilité : **84.7%**
- Base model trainable : **oui**
- Paramètres entraînables : **17,927,370**
- Temps d'entraînement total : **31 min**
- Taille de l'image : 224 x 224
- Train size / validation size / test size : 0.7/0.2/0.1
- Batch size : 32
- Learning rate : 0.00001
- Epochs : 100
- Dropout : 0.5
- Dernière couche Dense : 128

Au fil de mes essais et de mes lectures¹, j'ai trouvé des paramètres qui ont une grande importance et un grand impact sur la fiabilité du CNN :

24.3.3 Learning Rate

Contrôle le taux d'apprentissage : définit la rapidité avec laquelle les poids de l'algorithme d'optimisation sont mis à jour à chaque passage. Un taux bas est préférable pour nous.

- **Learning rate bas** : augmente le temps d'apprentissage mais converge plus aisément
- **Learning rate haut** : apprentissage accéléré en mettant drastiquement à jour les poids mais peut empêcher la convergence

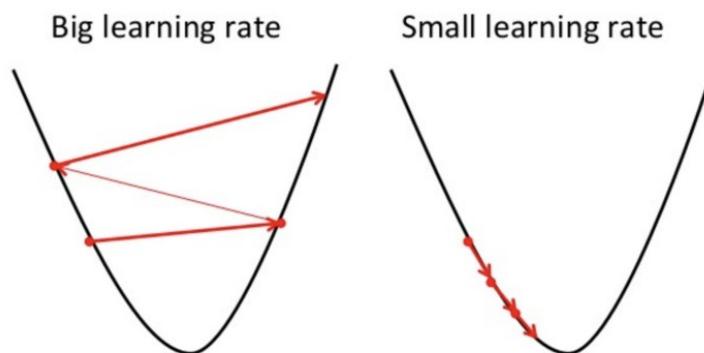


Figure 24.10 : Influence du learning rate sur le gradient descent

¹ <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>

24.3.4 Trainable Weights

On utilise des modèles de CNN existant pour faire du transfer learning. On peut, pour l'entraînement, choisir de verrouiller ou non les couches de ce CNN. Dans notre cas, rendre les couches du modèle CNN de base entraînables a drastiquement augmenté la fiabilité. Cela s'explique car il a été entraîné sur "ImageNet" qui ne correspond absolument pas à notre application.

- **Trainable weights** : on peut mettre à jour chaque couche du modèle existant en plus des couches que nous avons ajoutées.
- **Freeze layers** : utiliser les poids par défaut et exclure les couches du modèle CNN de l'entraînement. Entrainer uniquement les couches que nous avons ajoutées.

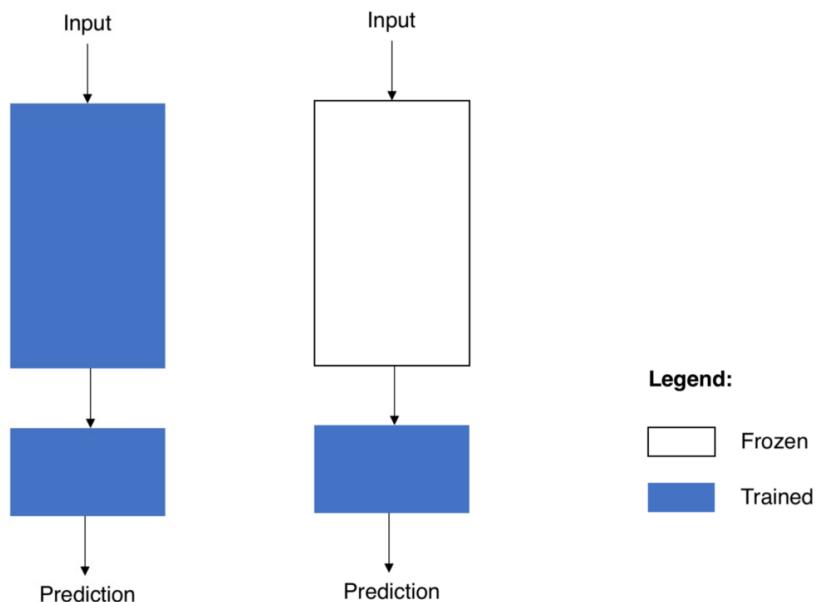


Figure 24.11 : Modèle avec poids de base entraînables et poids de base verrouillées¹

¹ <https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751>

24.3.5 Dropout¹

Le dropout est une technique de régularisation pour éviter l'overfitting dans les NN. Il supprime simplement certains neurones en fonction d'une probabilité donnée. Cela permet concrètement d'augmenter la fiabilité sur le set de validation et permet une meilleure généralisation du modèle.

Un dropout de 0.5 est optimal dans mon projet.

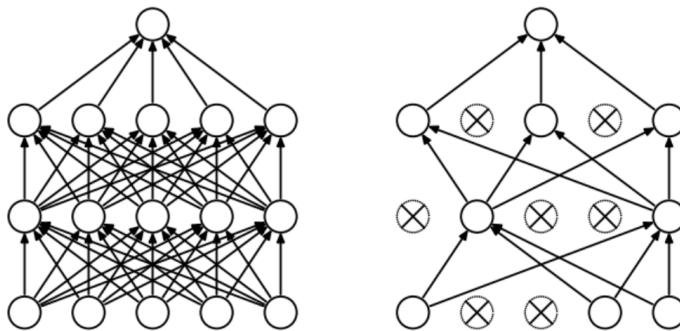


Figure 24.12 : Schématisation d'un NN sans dropout et avec dropout à 50%²

24.4 Performances avec dataset "balanced"

J'ai manuellement modifié le dataset Tobacco3482 afin qu'il soit balancé (que chaque classe aie le même nombre de samples). J'ai obtenu le dataset suivant : Total de **2'398** images que nous séparons en : **1'680** images d'entraînement, **480** images de validation et **238** images de test.

Le nombre d'images total étant plus petit, le temps d'entraînement sera forcément impacté positivement. Les performances ne sont cependant que faiblement dégradées avec seulement **0.4 % de moins**.

Dataset	Modèle	Divergence	Temps d'entraînement	Fiabilité
Tobacco3482	VGG-16	± 500	1'900 sec (31.6 min)	84.7 %
Tobacco3482 Balanced	VGG-16	± 0	1'300 sec (21.6 min)	84.0 %

¹ <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5>

² <http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf>

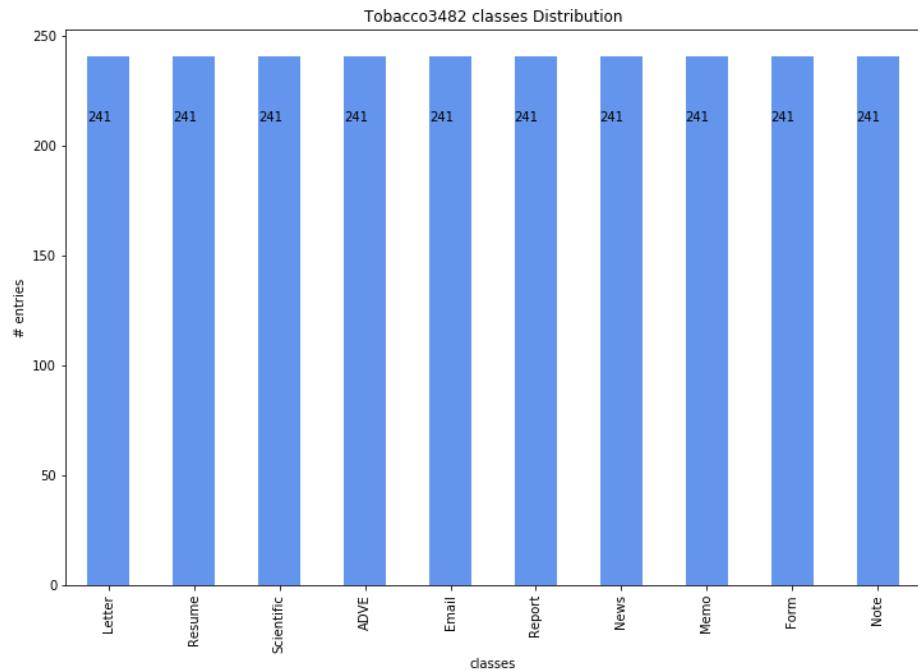


Figure 24.13 : Répartition des classes pour le dataset "balanced"

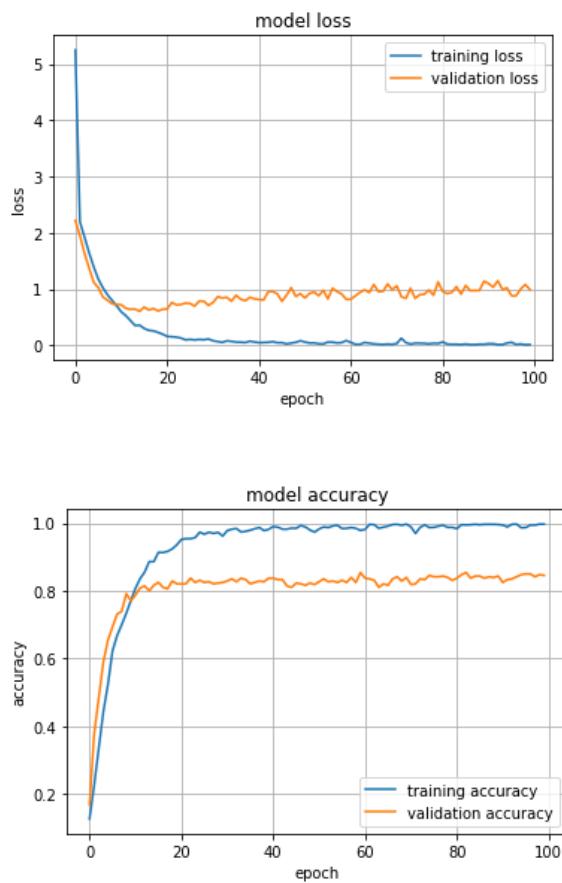


Figure 24.14 : Visualisation de l'évolution de la perte et de la fiabilité au fil des epochs

25 Performances NER

Le réentraînement n'ayant pas fonctionné, je suis dans l'incapacité de l'évaluée.

Nous avons cependant pu constater que la fiabilité des modèles de base n'est absolument pas suffisante et il est donc impossible de s'en servir tel quel.

26 Performances FSM

Pour le tester, j'ai introduit volontairement des erreurs qui pourraient être faites par l'OCR :

Mot de base : Medecin

Dérivés : MEDECIM, Médecln, Médccln

26.1 Evaluation des distances

Mot dérivé	Score Levenshtein	Score Jaro
MEDECIM	86	0.90
Médecln	77	0.74
Médccln	62	0.71

On peut constater que Jaro continue, visiblement, d'avoir une bonne fiabilité même si le mot subit de grosses modifications. Mais ce n'est pas si facile de tirer une conclusion, Jaro obtient également énormément de "match" incohérents autour des 75%, contrairement à Levenshtein qui reste très cohérent.

La raison de ces match "fantaisistes" est que Jaro, comme expliqué dans le chapitre Fuzzy String Matching, ne compte pas les opérations d'ajout et de suppression de caractère alors que pour Levenshtein, ce sont des opérations très couteuses. Jaro aura donc une forte tendance à interpréter des mots totalement différents ("audriaz" matché avec "a" par exemple) et à considérer leurs similitudes suffisantes.

Levenshtein sera donc la métrique qu'il faudra utiliser.

```
> jaro python find.py
('search : ', 'terieur')
found : Service
('score : ', 0.7142857142857143)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : MEDECIM
('score : ', 0.9047619047619048)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : Médeclin
('score : ', 0.7428571428571429)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : Médccln
('score : ', 0.7142857142857143)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'audriaz')
found : a
('score : ', 0.7142857142857143)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
```



```
> levenshtein python find.py
('search : ', 'audriaz')
found : Riaz
('score : ', 73)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'audriaz')
found : Riaz
('score : ', 73)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : MEDECIM
('score : ', 86)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
('search : ', 'medecin')
found : Médeclin
('score : ', 77)
('file : ', 'input/aaa/Classeur1 3.pdf.txt')
-----
```

Figure 26.1 : Visualisation des incohérences sorties par les deux méthodes

Synthèse

Ce chapitre nous a permis de tester, d'évaluer et de valider la performance du système développé. Principalement le CNN qui a, selon moi, atteint une fiabilité honorable. Les systèmes de NER sont complètement hors-course dans son état actuel et le Fuzzy String Matching a montré qu'il est utilisable dans notre champ d'application. Nous avons maintenant une bonne idée des compétences des différents "modules" développées et savons ce que nous devons améliorer afin de rendre le système déployable.

VIII. CONCLUSIONS

27 Conclusion du projet

27.1 Validation des objectifs

1. Reconnaître de manière autonome le type du document scanné

Tous les objectifs spécifiés dans cette partie ont été remplis. Un système de reconnaissance autonome de document avec une bonne fiabilité a été réalisé au moyen du deep learning.

2. Les regrouper s'il y a plusieurs pages

Cet objectif a été écarté lors du remaniement des objectifs à cause d'un manque de temps. La cause est un semestre 6 plus court que le 5 et la quantité de travail un peu sous-estimée de ma part. Cet objectif pourra bien sûr être rempli par un étudiant lors d'une suite potentielle du projet.

2. Extraire les informations qui y sont liées

L'objectif a été partiellement rempli, les techniques de NER et de FSM proposées dans la partie analyse technique ont été mis en œuvre et testés mais divers problèmes techniques technique n'ont pas permis de pousser à bout le réentraînement du NER.

Le manque de temps ne m'aura pas permis de résoudre ce problème avant le rendu du projet. Cette partie analyse de texte a été écarté en accord avec les professeurs responsables et l'accent a été mis sur l'analyse d'images pour cette même raison.

4. Faire une étude de marché, de la concurrence et de modèle économique

Tous les objectifs spécifiés dans cette partie ont été remplis. Une analyse cohérente et conséquente a été réalisée et permet à e-sculape de se positionner sur le marché.

27.2 Problèmes rencontrés et solutions apportées

27.2.1 Données de e-sculape non adaptées

Nous avons vu dans le chapitre : Confidentialité ci-dessus les raisons pourquoi les données (documents scannés) fournies par e-sculape ne sont pas adaptées à une approche d'analyse d'image. C'est pour pallier ces problèmes que nous avons choisi d'utiliser des dataset existants et dédiés à l'entraînement de classification de documents. Nous fournissons ainsi un modèle fonctionnel et testé afin de prouver que cette approche fonctionne. E-sculape pourront ensuite le ré-entraîner avec leurs propres documents et classes.

27.2.2 RVL-CDIP

Un des dataset proposé pour l'entraînement était RVL-CDIP. Je n'ai pas pu l'utiliser à cause de manque de place et de performances de ma machine. En effet, le fichier en `.tar.gzip` fait déjà 37 GB, décompressé, sa taille atteint les 70 GB. De plus, avec 400'000 images en son sein, les temps nécessaires pour y faire des actions deviennent déraisonnables.

27.2.3 Transfer Learning avec Keras

N'étant pas familier avec Python et n'ayant jamais touché au machine learning avant ce projet, j'ai fait face à de nombreux obstacles que j'ai dû franchir. Un gros travail d'apprentissage en autodidacte a été nécessaire. Ma persévérance ainsi que de nombreuses heures passées à débugger, surfer sur les forums et me documenter m'ont permis de venir à bout du projet en ayant acquis un grand bagage théorique et pratique.

27.2.4 Analyse de texte

Comme déjà mentionné ci-dessus, la phase d'analyse de texte a été un peu mise de côté en faveur de l'analyse d'image. Cette décision a été prise conjointement avec mes responsables de projet par cause de manque de temps.

27.2.5 Mauvaise compréhension du DEP

N'étant pas du tout du domaine médical, il m'a fallu quelques semaines afin de bien comprendre les interactions ainsi que le fonctionnement de tout le système de santé numérique Suisse. M. Clément m'a été d'une grande aide pour me fournir un appui dans ce domaine que je ne maîtrisais pas et m'a permis de fournir une analyse cohérente.

27.2.6 Organisation

La semaine avant les vacances de Pâques a engendré une prise de retard sur le planning à cause de nombreux examens. J'en ai discuté avec les responsables et leur ai assuré que le retard pris serait rattrapé durant la semaine de vacances. Tout s'est déroulé ensuite comme prévu et le retard a été amorti.

27.3 Perspectives futures

27.3.1 Regrouper les documents

Les documents donnés en input au modèle sont reconnus et classifiées grâce au machine learning mais certains documents possèdent plusieurs pages. Pour le moment, aucun réassemblage de ces documents est fait. Un document scanné est égal à un fichier PDF alors qu'il devrait y avoir des fichiers PDF de plusieurs pages contenant toutes les pages d'un même document.

27.3.2 Triplet Loss

Nous avons vu qu'il est nécessaire, dans une approche avec de CNN, de fournir beaucoup de données durant la phase d'entraînement afin d'espérer avoir une fiabilité satisfaisante. On pourrait se pencher sur d'autres algorithmes d'apprentissage en nécessitant beaucoup moins comme les méthodes dites "one-shot learning" comme le Triplet Loss par exemple.

27.3.3 NLP pour reconnaître le type de document

Nous utilisons une méthode basée sur l'analyse d'image pour reconnaître le type de document, mais vu que nous possédons des données d'OCR, il serait envisageable de les utiliser afin de retrouver le type de document au moyen de méthodes d'apprentissages basées sur l'analyse de texte (NLP).

27.3.4 NER

Avoir un système de NER qui soit ré-entraîné et fiable sur nos sets de données serait également un vrai plus pour le projet et un travail intéressant à faire.

27.3.5 RVL-CDIP

Si un ordinateur puissant est mis à disposition, il est envisageable de relancer les diverses phases d'entraînement sur les CNN avec ce dataset et de comparer avec Tobacco3482.

27.3.6 Intégration de la solution

Dans les activités optionnelles, il est proposé d'intégrer la solution développée dans le cadre du projet aux installations de e-sculape. Le but est d'avoir une application intégrant de manière harmonieuse notre solution et étant prêt à être commercialisée

27.4 Remerciements

Je tiens à remercier du fond du cœur les personnes qui m'ont encadré, renseigné, soutenu et aidé tout au long de ce projet :

- **Monsieur Nicolas Schroeter et Monsieur Andreas Fischer**, professeurs à la HEIA-FR, pour leur suivi et leur supervision en tant que responsables de projet durant l'entier du semestre.
- **Monsieur Jérôme Clément**, fondateur de l'entreprise d'informatique médical e-sculape pour avoir proposé ce projet et avoir mis à disposition ses connaissances dans le domaine.

Merci pour leur précieuses remarques et informations qui m'ont permis d'améliorer mon travail et d'en être fier.

27.5 Conclusion du projet

Comme indiqué ci-dessus, les objectifs principaux du cahier des charges ont été remplis et le projet est, selon moi, une réussite. J'ai créé un prototype et prouvé qu'une reconnaissance autonome du type d'un document peut être fait de manière fiable au moyen de techniques de Deep Learning. J'ai également fourni une analyse économique conforme aux demandes du mandant et lui permettant de positionner son produit sur le marché.

Il est bien sûr frustrant de ne pas avoir pu étudier plus en profondeur les méthodes de NLP et de NER, les méthodes de one-shot learning et de ne pas avoir pu intégrer les modules développés entre eux. Mais le temps est malheureusement compté et il faut savoir conclure.

Je suis cependant confiant que ce projet et ce rapport constituent une base solide pour quiconque souhaite en reprendre le flambeau et finaliser le long voyage qui a été entrepris.

27.6 Conclusion personnelle

J'ai été attiré très vite par ce projet car il constituait une aubaine pour moi de m'atteler au domaine du machine learning qui me rendait plus que curieux depuis quelque temps déjà. J'ai profité de cette porte d'entrée pour m'y mettre et ne regrette pas une seule seconde d'avoir eu le "courage" de lancer dans ce domaine si vaste dont je ne connaissais quasiment rien. Cela a occasionné de nombreux questionnements et problèmes liés intimement à ma méconnaissance sur sujet mais cela m'a forcé à sortir hors de ma zone de confort et à faire un vrai effort d'apprentissage en autodidacte. Je ressors mûri de cette expérience et me suis découvert un vrai intérêt pour ce domaine en plein essor. La preuve, il sera également au cœur de mon projet de bachelor.

Il fut également intéressant et enrichissant de travailler en collaboration avec une entreprise même si j'ai eu de la peine à les tenir au courant de l'avancée sur le projet, étant plus concentré sur l'avancement pur de celui-ci selon le cahier des charges que sur le feedback que pourrait m'apporter le mandant.

Le semestre 6 étant excessivement court, la dose de travail requise par ce projet de semestre fut parfois pénible et aura engendré passablement d'anxiété en moi. J'ai eu un peu de mal à m'y mettre à fond au départ. Le projet étant assez flou, le potentiel non-usage du machine learning et une phase d'analyse très longue ont miné ma motivation initiale. Heureusement, les nuages se sont vite dissipés et je me suis retrouvé un entrain inconsidéré durant la réalisation, n'arrivant plus lâcher mon clavier et enchaînant les heures de travail sans vergogne afin de gratter chaque petit pourcent de fiabilité en plus de mon système Deep Learning.

Je ressors grandit de ce second projet de semestre. Moi, Patrick, suis fier de vous présenter le fruit de mon travail.

28 Conclusion du document

28.1 Licences

Software	Version	License
Keras	2.2.3	MIT
OpenCV	4.0.1	3-clause BSD License
Matplotlib	3.0.3	PSF
Pandas	0.24.2	BSD 3-Clause License
Numpy	1.16.1	BSD
Pdftotext	4.01	MIT
ImageMagick	7.0.8	Apache 2.0

28.2 Déclaration d'honneur

Je, soussigné, Patrick Audriaz, déclare sur l'honneur que le travail rendu est le fruit d'un travail personnel. Je certifie ne pas avoir eu recours au plagiat ou à toutes autres formes de fraudes. Toutes les sources d'information utilisées et les citations d'auteur ont été clairement mentionnées.

Fribourg, 10 mai 2019

Patrick Audriaz

IX. REFERENCES

1. **(OFSP), Office fédéral de la santé publique.** Stratégie Cybersanté Suisse 2.0. www.admin.ch. [En ligne] 14 décembre 2018. [Citation : 25 avril 2019.] <https://www.bag.admin.ch/bag/fr/home/strategie-und-politik/nationale-gesundheitsstrategien/strategie-ehealth-schweiz.html>.
2. —. Législation Dossier électronique du patient. www.admin.ch. [En ligne] 8 mars 2019. [Citation : 25 avril 2019.] <https://www.bag.admin.ch/bag/fr/home/gesetze-und-bewilligungen/gesetzgebung/gesetzgebung-mensch-gesundheit/gesetzgebung-elektronisches-patientendossier.html>.
3. **Frehner, Ludovic.** *Projet – Scanning*. E-sculape. 2017. Analyse préliminaire.
4. **Beat Heggli, Groupe d'utilisateurs HL7 Suisse.** *CDA-CH-II: Spécification pour la création de modèles Health level 7 clinical document architecture*. 2011.
5. **Van Tien Nguyen, Christian Sallaberry, Mauro Gaio.** *Mesure de la similarité entre termes et labels de concepts ontologiques*. Neuchâtel, Suisse : archives-ouvertes.fr, 2013. hal00847528.
6. **Andreas Kolsch, Muhammad Zeshan Afzal, Markus Ebbecke, Marcus Liwicki.** *Real-Time Document Image Classification using Deep CNN and Extreme Learning Machines*. 2017. arXiv:1711.05862v1.
7. **Arindam Das, Saikat Roy, Ujjwal Bhattacharya, Swapan K. Parui.** *Document Image Classification with Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Neural Networks*. Beijing, China : s.n., 2018. 978-1-5386-3788-3.
8. **Géron, Aurélien.** *Hands on Machine Learning with Scikit Learn and TensorFlow*. s.l. : O'Reilly, 2017. 978-1-491-96229-9.
9. **Burkov's, Andriy.** *The Hundred-Page Machine Learning Book*. 2018.
10. **Buduma, Nikhil.** *Fundamentals of Deep Learning*. s.l. : O'Reilly, 2017. 978-1-491-92561-4.
11. **Jayant Kumar, Peng Ye, David Doermann.** *Structural Similarity for Document Image Classification and Retrieval (Tovacco3482 Dataset)*. 2013.
12. **Muhammad Zeshan Afzal, Andreas Kölsch, Sheraz Ahmed, Marcus Liwicki.** *Cutting the Error by Half: Investigation of Very Deep CNN and Advanced Training Strategies for Document Image Classification*. 2017. 1704.03557v1.

13. **Elad Hoffer, Nir Ailon.** *DEEP METRIC LEARNING USING TRIPLET NETWORK*. Israel : s.n., 2018. 1412.6622v4.
14. **Ariel Gordon, Elad Eban, Ofir Nachum, Bo CHen, Hao Wu, Tien-Ju Yang, Edward Choi.** *MorphNet: Fast & Simple Resource-Constrained Structure Learning of Deep Networks*. 2018. arXiv:1711.06798v3.
15. **Chollet, François.** *Deep Learning With Python*. s.l. : Manning Publications Co., 2018. ISBN 9781617294433.

X. GLOSSAIRE

CNN : Convolutional Neural Network

DB : Database (base de données)

DEP / EPD : Dossier électronique du patient

DL : Deep Learning

ERP : Enterprise Resource Planning ou progiciel de gestion intégré

FSM : Fuzzy String Matching

JSON : JavaScript Object Notation

LDEP : Loi fédérale sur le dossier électronique du patient

ML : Machine Learning

NER : Named Entity Recognition

NLP : Natural Language Processing

NN : Neural Network

OCR : Reconnaissance optique de caractères

PDF : Portable Document Format

TL : Transfer Learning

XML : eXtensible Markup Language

XI. TABLE DES FIGURES

Figure 8.1 : Logo du DEP	17
Figure 8.2 : Schéma de l'architecture pour la mise en place du DEP.....	18
Figure 9.1 : Critères d'aide au choix d'un cible.....	24
Figure 9.2 : Canvas de proposition de valeur.....	25
Figure 9.3 : Modélisation du canvas de proposition de valeur pour notre projet (e-sculape en haut, cabinets médicaux en bas).....	26
Figure 9.4 : Analyse SWOT.....	31
Figure 10.1 : Logiciel pour entrer les informations du patient et imprimer les codes QR.....	34
Figure 10.2 : Exemples d'autocollants de code QR générées	34
Figure 10.3 : Codes QR collé sur les documents en fonction de leur type.	35
Figure 10.4 : Exemple de fichier PDF sortie avec OCR.....	36
Figure 10.5 : En-tête de l'XML (CDA-CH) généré.....	36
Figure 10.6 : Encapsulation du PDF en base64 dans le XML.....	36
Figure 10.7 : Documents scannés et données associées disponibles sur l'ERP	37
Figure 10.8 : Schéma du workflow actuel	37
Figure 10.9 : Schématisation du nouveau workflow hypothétique	38
Figure 11.1 : Diagramme du flux des données actuel	39
Figure 11.2 : Diagramme de classe de la base de données.....	40
Figure 11.3 : Exemple d'entrée de la base de données (médecin) exportée en JSON	40
Figure 12.1 : Exemple d'erreur avec le PDF à gauche et l'OCR à droite....	41
Figure 12.2 : Pour récupérer l'OCR d'un PDF sous forme de texte	41
Figure 12.3 : Exemple d'anonymisation et de masquage.....	42
Figure 13.1 : Exemple de structure XML de document CDA-CH.....	43
Figure 14.1 : Exemple des opérations primaires.....	45
Figure 14.2 : Eventail des différentes méthodes de String Matching (5) ...	46
Figure 14.3 : Exemple de résultat de NER.....	50
Figure 14.4 : Schéma d'apprentissage supervisé pour de la classification, phase d'entraînement et de prédiction	50
Figure 14.5 : Exemples d'entités reconnues par le modèle pré-entraîné de spaCy	51
Figure 14.6 : Exemple d'utilisation de spaCy qui ressort les informations intéressantes d'une phrase	51

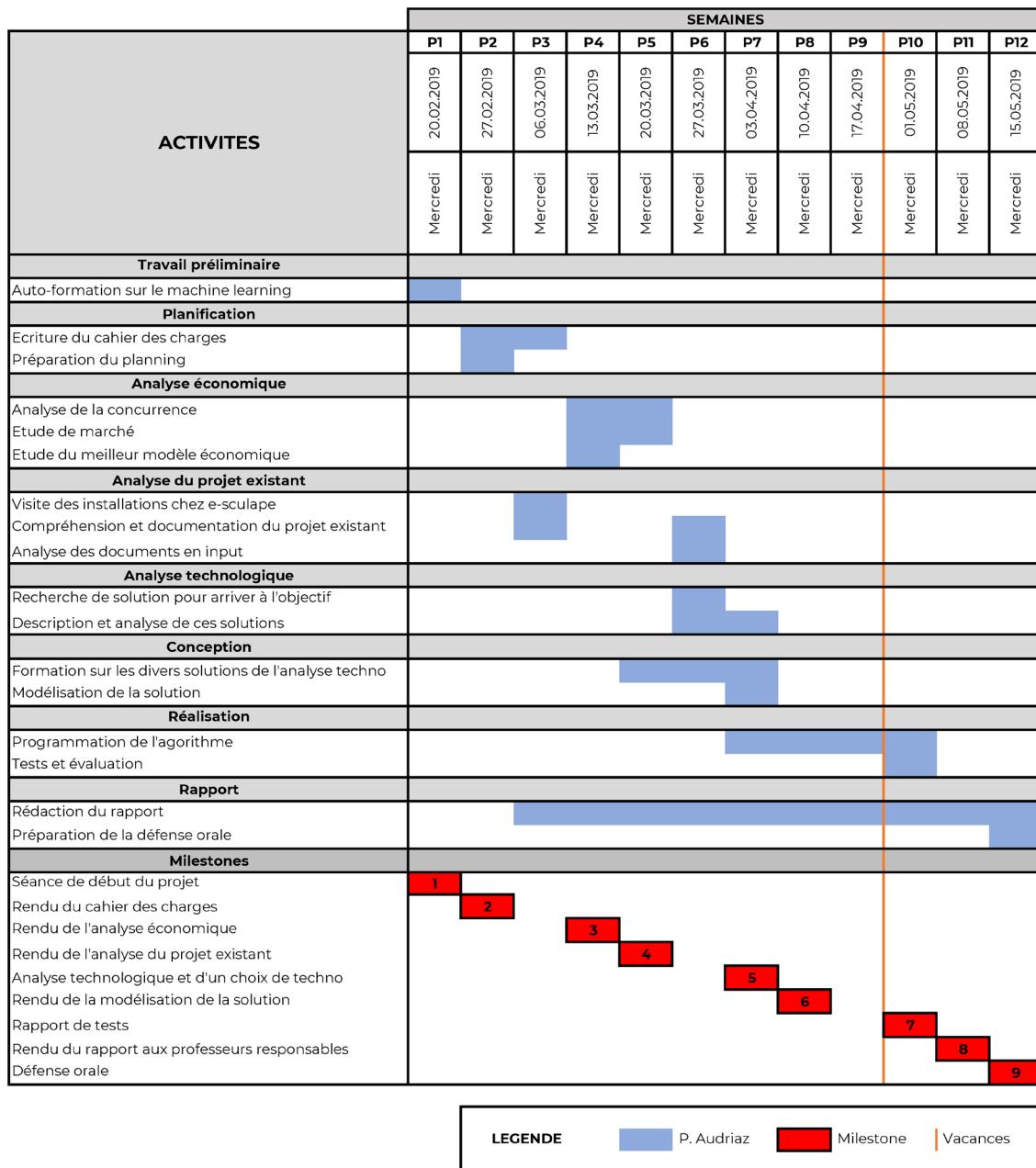
Figure 14.7 : NER par le Stanford Named Entity Recognizer.....	52
Figure 15.1 : Visualisation de la différence entre ML et DL	55
Figure 15.2 : Schématisation du fonctionnement d'un modèle CNN.....	56
Figure 15.3 : Extractions de features par un algorithme CNN sur un visage humain	56
Figure 15.4 : Schématisation du transfer learning	57
Figure 15.5 : Comparaison des différentes architectures CNN	59
Figure 15.6 : Classes de documents et exemple pour chaque du RVL-CDIP Dataset.....	60
Figure 15.7 : Répartition des classes pour le dataset Tobacco3482.....	61
Figure 15.8 : Précision obtenue sur le dataset Tobacoo3482 en fonction de diverses méthodes de transfer learning et différents modèles de CNN (12)	61
Figure 15.9 : Schématisation du fonctionnement de l'apprentissage via Triplet Loss	63
Figure 15.10 : Visualisation de la distribution des données avant (gauche) et après (droite) l'apprentissage d'un Triplet Network	63
Figure 16.1 : Structure du dataset avec les dossiers labelisés contenant les images.....	65
Figure 16.2 : Structure de fichiers utilisé par le CNN pour l'entraînement (fait de manière autonome par l'algorithme).....	66
Figure 16.3 : Schéma du workflow d'entraînement du modèle.....	66
Figure 16.4 : Schéma de la nouvelle architecture qui va être mise en place	68
Figure 17.1 : Exemple de Notebook Jupyter montrant du texte formaté, un bloc de code, la console avec le résultat et une figure.....	69
Figure 18.1 : Evolution de l'intérêt pour différents framework	70
Figure 18.2 : Keras Workflow afin de créer, entraîner et utiliser un modèle	70
Figure 19.1 : Script bash utilisant pdftotext pour faire la batch conversion	71
Figure 19.2 : Exemple de document PDF scanné.....	71
Figure 19.3 : Résultat obtenu sur le même document après extraction de l'OCR en format texte	72
Figure 20.1 : Script bash utilisant convert pour faire la batch conversion	72
Figure 21.1 : Schématisation de la division du dataset et différents sous-sets	74
Figure 21.2 : Distribution des données dans les différents sets.....	76

Figure 21.3 : Fonction utilisant opencv (cv2) pour redimensionner les images	77
Figure 21.4 : Document avant et après redimensionnement	77
Figure 21.5 : transforme la liste d'images et de labels en Numpy Array, print leur "forme" ainsi que la valeur des pixels	78
Figure 21.6 : Binarisation des labels : print avant et après transformation et print des classes contenues dans le binarizer	79
Figure 21.7 : Création, compilation en entrainement d'un modèle VGG-16	81
Figure 21.8 : Structure du modèle crée, on y retrouve toutes les couches que nous avons défini	83
Figure 21.9 : Création d'un modèle ResNet50	83
Figure 21.10 : Création d'un modèle InceptionV3	84
Figure 21.11 : Création d'un modèle Xception	84
Figure 21.12 : Test de la fiabilité (accuracy) et des erreurs (loss)	84
Figure 21.13 : Evolution de l'accuracy et du loss au fil des epochs	85
Figure 21.14 : Sauver le modèle	86
Figure 22.1 : Script Python utilisant SpaCy pour faire du NER en batch sur l'OCR en batch et sauvegarder le résultat en JSON	87
Figure 22.2 : Résultat du NER français sur le texte de l'OCR	87
Figure 22.3 : Output sous format JSON avec la classe comme clé et les valeurs correspondantes	88
Figure 22.4 : Visualisation du résultat du NER en anglais sur le texte de l'OCR	88
Figure 22.5 : Exemple de fichier JSON contenant les données texte et leur classe attribuée (label)	90
Figure 22.6 : Commandes à exécuter dans le terminal afin d'entrainer le modèle	90
Figure 22.7 : Erreur retournée	91
Figure 23.1 : Exemple d'entrées dans la DB du patient et du médecin au format JSON contenant les entrées que nous pourrons matcher avec les mots de l'OCR	92
Figure 23.2 : Utilisation de la librairie pour trouver les matchs au moyen de la distance de Levenshtein	92
Figure 23.3 : Résultat obtenu en recherchant le nom des médecins de la DB dans l'OCR des fichiers données en entrée avec affichage de la distance (score)	93
Figure 23.4 : Utilisation de la librairie pour trouver les matchs au moyen de la distance de Jaro	93

Figure 23.5 : Résultat obtenu en recherchant le nom des médecins de la DB dans l'OCR des fichiers données en entrée avec affichage de la distance (score)	94
Figure 24.1 : VGG-16 Figure 24.2 : ResNet-50.....	95
Figure 24.3 : InceptionV3Figure 24.4 :	Xception
	96
Figure 24.5 : Schématisation de courbes qui indiquent un overfitting du modèle, notre modèle n'overfit pas.....	97
Figure 24.6 : Evaluation du learning rate en fonction de la forme de la courbe, le nôtre est adéquat.....	97
Figure 24.7 : "Deep learning is an art, not a science".....	98
Figure 24.8 : Graphique d'évolution de la fiabilité et des erreurs au fil du temps.....	99
Figure 24.9 : Graphique d'évolution de la fiabilité et des erreurs au fil du temp.....	99
Figure 24.10 : Influence du learning rate sur le gradient descent.....	100
Figure 24.11 : Modèle avec poids de base entraînables et poids de base verrouillées.....	101
Figure 24.12 : Schématisation d'un NN sans dropout et avec dropout à 50%.....	102
Figure 24.13 : Répartition des classes pour le dataset "balanced".....	103
Figure 24.14 : Visualisation de l'évolution de la perte et de la fiabilité au fil des epochs	104
Figure 26.1 : Visualisation des incohérences sorties par les deux méthodes	105

XII. ANNEXES

1 Planning



2 Kaggle Jupyter Notebook (CNN-16)

Ci-après :