



UNIVERSITÉ
LAVAL

FACULTÉ DE SCIENCES ET GÉNIE

CLASSIFICATION D'IMAGES RADIOLOGIQUES DE PNEUMONIE

INTRODUCTION À L'APPRENTISSAGE MACHINE (GIF-4101)
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE

ISAAC NERI GOMEZ SARMIENTO
111 274 808

DENIZ SOYSAL
111 274 472

STÉPHANE VOYER-BOUTET
908 201 709

FLORIAN PHILIPPE
111 136 789

18 DÉCEMBRE 2019

I. PRÉSENTATION DU PROBLÈME ET L'ÉTAT DE L'ART

Plusieurs milliers de personnes meurent au Canada, victimes de maladies pulmonaires telles que la grippe et la pneumonie. Un moyen de détecter la pneumonie est via l'imagerie à rayon X. Un jeu de données contenant des images classifiées est disponible sur le site Kaggle [1]. La figure 1 est un exemple du jeu de données. Ce dernier nous a permis d'expérimenter plusieurs approches d'apprentissage machine pour en trouver une qui maximisera le taux de détection de la pneumonie. L'apprentissage machine est un outil qui permet potentiellement d'améliorer, entre autres, la détection des problèmes de santé de plus en plus tôt et, avec le temps et l'amélioration des technologies, de manières plus certaines. Il existe désormais des robots capables de diagnostiquer des maladies en mesurant plusieurs indicateurs (variables) d'un patient. Dans le cadre de ce projet, notre but est de permettre la détection prématurée des pneumonies afin d'aider les professionnels de santé dans leur travail et ainsi améliorer l'espérance de vie des patients atteints.

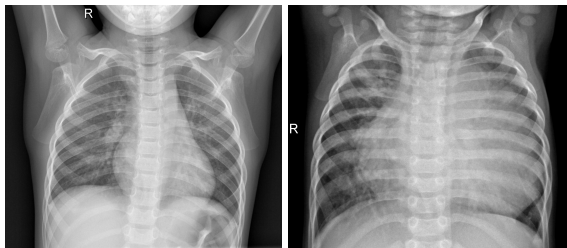


FIGURE 1 – L'image à gauche est un patient normal et l'image à droite est un patient avec pneumonie.

Concernant les connaissances acquises dans ce domaine, il est difficile de connaître la précision de détection de maladie pulmonaire comme la pneumonie. En effet, l'imagerie à rayons X est un outil parmi d'autres qui aide au diagnostic de la maladie. L'imagerie complète la détection de différents symptômes établis via une prise de sang, etc. Enfin, les professionnels de santé n'ont pas tous la même expérience dans le domaine, certains posent un diagnostic plus précis que d'autres, reconnaissent quand il y a une maladie pulmonaire. Cependant, d'après plusieurs études, le taux de détection devrait se situer entre 85 et 95%. [2] La récente démocratisation de l'apprentissage machine a entraîné un fort intérêt pour la conception d'algorithmes appliqués à ce type de

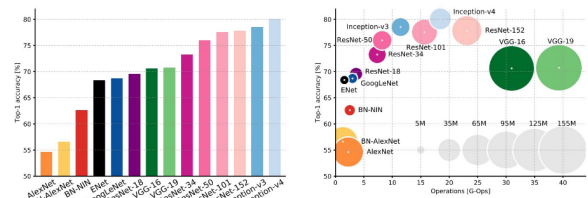
problème. Ainsi, plusieurs approches ont été documentées, ce qui a permis à notre équipe de choisir une approche qui sera développée plus bas dans le présent rapport.

II. APPROCHE PROPOSÉE

Après diverses recherches, nous avons finalement décidé d'utiliser l'approche d'apprentissage par transfert. [2] Les avantages de l'apprentissage par transfert par rapport à un apprentissage de zéro sont :

- Méthode très efficace sur un jeu de données relativement petit. Dans notre cas, nous avons $N = 5856$.
- Prend beaucoup moins de temps et est beaucoup moins lourd.
- Plus simple à implémenter.

Ensuite, il faut choisir le modèle à utiliser.



An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Source : https://miro.medium.com/max/1150/1*ZqkLRkMU20b0QWHLBg8sw.png

FIGURE 2 – Comparaison de différents modèles [3]

Nous avons retenu 3 modèles, que nous allons comparer en matière de précision et de vitesse d'exécution sur notre problème afin de déterminer le plus performant :

- VGG-16
- ResNet-101
- DenseNet-161

III. MÉTHODOLOGIE EXPÉRIMENTALE POUR ANALYSE ET COMPARAISON DES RÉSULTATS

Afin de choisir le modèle le plus performant, nous avons entraîné chacun de nos modèles avec 10 époques en faisant varier le taux d'apprentissage pour 10, 1, 0.1, 0.01 et 0.001. Pour les modèles VGG16 et DenseNet161, la taille d'images utilisée était 128x128, 256x256, alors que pour le ResNet101, nous avons utilisé les mêmes tailles et en plus 512x512. On a fixé le batch size à 32 et le momentum à 0.9. On a utilisé comme optimiseur la Descente du Gradient Stochastique et comme fonction de perte la Cross Entropy Loss. Pour les 3 différentes architectures, nous avons généré des graphiques qui comparent la précision de la validation en fonction du nombre d'époques, la fonction de perte en fonction du nombre d'époques et la précision de la validation et de l'entraînement en fonction des époques. Nous avons finalement généré un tableau reprenant la précision et le temps d'exécution des 3 architectures sur notre problème. Nous avons choisi l'architecture qui a eu la meilleure précision avec les paramètres optimaux. Avec cette architecture, nous avons entraîné le modèle avec 60 époques, des images 256x256 et un taux d'apprentissage de 0,01.

IV. RÉSULTATS EXPÉRIMENTAUX

Afin de trouver les hyperparamètres qui permettent d'obtenir une bonne précision pour chacune des trois architectures proposées, nous avons entraîné des modèles différents en faisant varier la taille des images entre 128x128 pixels, 256x256 pixels et 512x512 pixels et en faisant varier le taux d'apprentissage entre 0.001, 0.01, 0.1, 1, et 10. Lors des tests, il s'est avéré que les images de 512x512 pixels étaient trop grandes pour la mémoire du processeur graphique utilisé pour les architectures VGG-16 et DenseNet-161. Cette taille d'image n'a donc pas été testée avec ces architectures.

Les précisions maximales obtenues pour 10 époques pour chaque variation d'hyperparamètres sont présentées à la Table 1 pour l'architecture ResNet-101, à la Table 2 pour l'architecture VGG-16 et à la Table 3 pour l'architecture DenseNet-161.

D'après les résultats obtenus, c'est le modèle ResNet-101 avec des images de 256x256 pixels et un taux

d'apprentissage de 0.1 qui donne les meilleurs résultats. Par contre, en inspectant la valeur de la fonction de perte à chaque époque, on peut voir que celle-ci a un comportement instable et le taux d'apprentissage semble trop grand pour que la descente de gradient converge vers un minimum. Ainsi, il a été décidé de garder l'architecture ResNet-101 avec des images de 256x256 pixels, mais en diminuant légèrement le taux d'apprentissage à 0.01.

TAB. 1 – Recherche en grille des meilleurs hyperparamètres pour le modèle ResNet-101

Taille (px) (%)	Taux app.	Précision max (%)
128	0,001	76,39
	0,01	81,94
	0,1	76,85
	1	82,41
	10	81,02
256	0,001	75,46
	0,01	83,80
	0,1	87,5
	1	81,48
	10	83,80
512	0.001	77,31
	0,01	81,48
	0,1	85,19
	1	87,04
	10	84,72

TAB. 2 – Recherche en grille des meilleurs hyperparamètres pour le modèle VGG-16

Taille (px)	Taux app.	Précision max (%)
128	0,001	75,46
	0,01	83,33
	0,1	50,00
	1	50,00
	10	50,00
256	0.001	79,17
	0,01	84,72
	0,1	50,00
	1	50,00
	10	50,00

TAB. 3 – Recherche en grille des meilleurs hyperparamètres pour le modèle DenseNet-161

Taille (px) (%)	Taux app.	Précision max (%)
128	0,001	76,39
	0,01	80,09
	0,1	80,56
	1	80,09
	10	50,00
256	0.001	81,94
	0,01	80,56
	0,1	76,85
	1	83,33
	10	50,00

Une fois l'architecture et les hyperparamètres choisis, nous avons entraîné un modèle ResNet-101 pour 60 époques afin de vérifier s'il serait possible d'obtenir un meilleur résultat avec plus d'époques que les 10 qui ont été testées précédemment. Nous avons noté que la précision du jeu de validation atteint son maximum après environ 9 époques (voir Figure 3). Sur cette même figure, on peut voir que la précision du jeu de validation diminue ensuite à cause du sur-apprentissage du jeu d'entraînement après environ 15 époques. On constate également que plus il y a d'époques, plus la valeur retournée par la fonction de perte diminue (voir Figure 4), ce qui indique que le modèle est de plus en plus en mesure de bien classer les données du jeu d'entraînement.

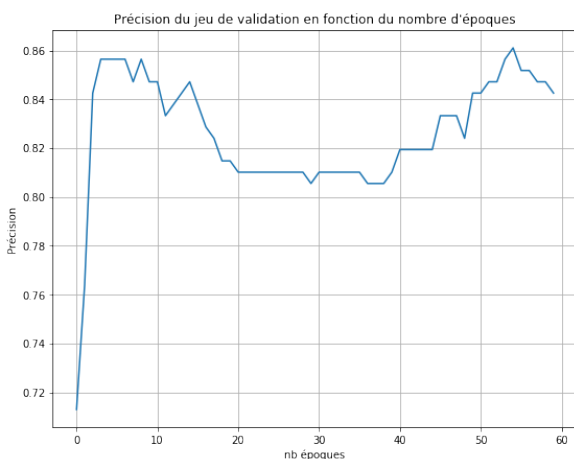


FIGURE 3 – Précision du modèle ResNet-101 avec le jeu de validation lors de l'entraînement avec 60 époques

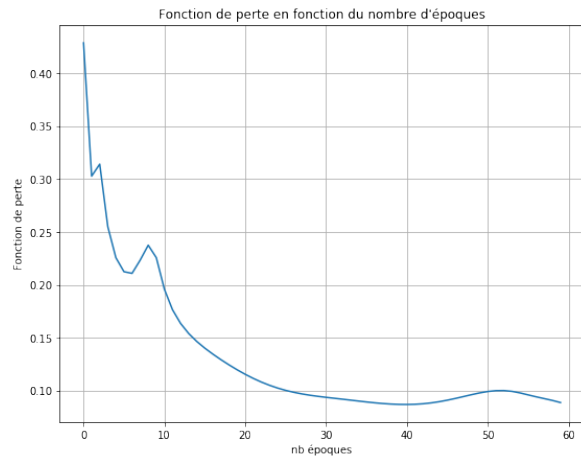


FIGURE 4 – Valeur de la fonction de perte du modèle ResNet-101 avec le jeu de validation lors de l'entraînement avec 60 époques

V. ANALYSE DES RÉSULTATS ET DISCUSSIONS

Après avoir identifié les hyperparamètres qui permettent d'atteindre une bonne précision avec le jeu de validation, nous avons entraîné un modèle ResNet-101 avec 9 époques, ce qui permet d'éviter le surapprentissage. La précision obtenue pour chaque époque est présentée à la Figure 5. On obtient pour le modèle final une précision de 83,80% sur le jeu de validation. La précision obtenue avec le jeu d'entraînement est de 92,31%. La dernière étape est de vérifier la précision obtenue avec le jeu de données qui n'est pas intervenu dans l'entraînement du modèle ; le jeu de test. On obtient une précision avec le jeu de test de 89,62%, ce qui indique la bonne précision à laquelle on peut s'attendre en catégorisant d'autres images avec ce modèle.

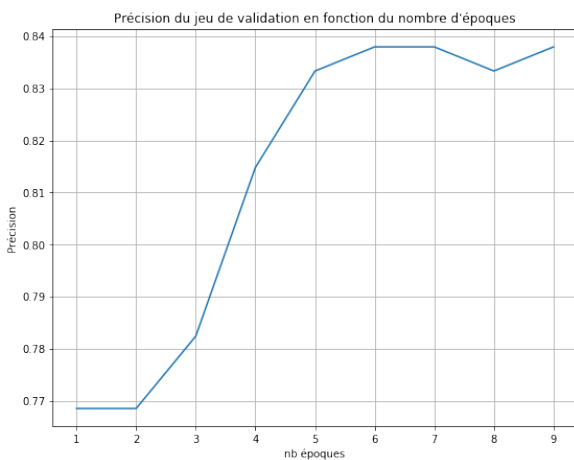


FIGURE 5 – Précision du modèle ResNet-101 à chaque époque avec le jeu de test lors de l'entraînement du modèle final

Nous avons calculé la matrice de confusion afin de voir quelles données, parmi celles mal classées, sont des faux positifs et de faux négatifs (voir Figure 6). On voit que 6,60% sont des faux positifs et 3,77% sont des faux négatifs. Ceci est dû à la plus grande proportion d'images de pneumonies dans le jeu d'entraînement. Pour éviter cela, il aurait été possible de balancer le jeu d'entraînement au départ. Par contre, pour ce problème particulier, nous considérons que les conséquences d'un faux négatif sont plus graves que celles d'un faux positif.

		Prédictions	
		Normal (0)	Pneumonie (1)
Classes	Normal (0)	106	28
	Pneumonie (1)	16	274

FIGURE 6 – Matrice de confusion de la classification du jeu de test

RÉFÉRENCES

- [1] <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>
- [2] [https://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS0092867418301545%3Fshowall%3Dtrue](https://www.cell.com/cell/fulltext/S0092-8674(18)30154-5?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS0092867418301545%3Fshowall%3Dtrue)
- [3] https://miro.medium.com/max/1150/1*ZqkLRkMU20b0QWIHLBg8sw.png
- [4] <https://vision.gel.ulaval.ca/~cgagne/enseignement/apprentissage/A2019/#travaux>