

Rapport de projet IN52/54

Détection de pneumonie



Automne 2018

Marine COLLET – Pierre CARCELLER – Bastien LUCHIER – Guillaume VANSON – Cédric PAQUET

Table des matières

Introduction	3
La pneumonie.....	4
Traitement de l'image et analyse de contours	5
Méthode.....	5
Redimensionnement	5
Egalisation d'histogramme	5
Recadrage.....	6
Identification	6
Seuillage d'Otsu.....	7
Combinaison et analyse.....	8
Limites et améliorations	8
Résultats.....	8
Réseau de neurones simple	9
Méthode.....	9
Analyse des données	9
Data-Augmentation.....	10
Pré-traitement.....	10
Architecture du réseau	11
Entraînement	11
Résultats.....	11
Architecture poussée	12
Conclusion.....	12

Introduction

Afin de mettre en application les compétences acquises dans les deux modules IN52 et IN54 nous avons décidé de travailler sur la détection de pneumonie chez les enfants à partir de radiographies du thorax.

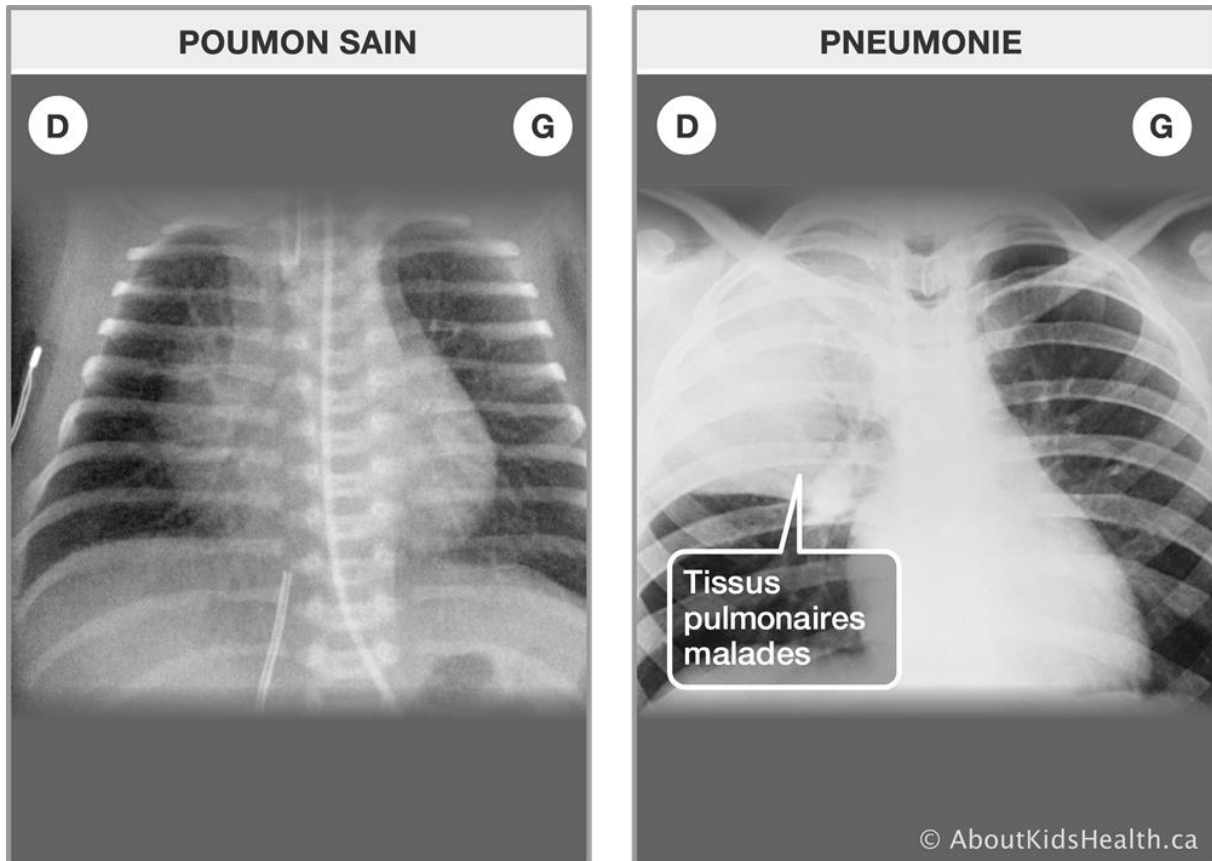
Ce projet est divisé en trois axes, tout d'abord la proposition d'une solution basée sur des algorithmes « traditionnels » d'imagerie. Ensuite, nous proposerons une solution basée sur les réseaux de neurones convolutifs en utilisant une architecture assez simple, puis dans un dernier temps, nous aborderons une piste nécessitant une architecture beaucoup plus lourde.

La pneumonie

La pneumonie est une infection des poumons causée le plus souvent par un virus ou une bactérie. L'infection touche plus précisément les alvéoles pulmonaires, ces minuscules sacs en forme de ballons situés à l'extrémité des bronchioles (voir le schéma).

Cette infection respiratoire est responsable de 15% des décès d'enfants âgés de moins de 5 ans. Cela représente 935 000 enfants morts de pneumonie en 2013 dans le monde.

La pneumonie est visible sur les radiologies du thorax pour les experts ayant un œil averti :

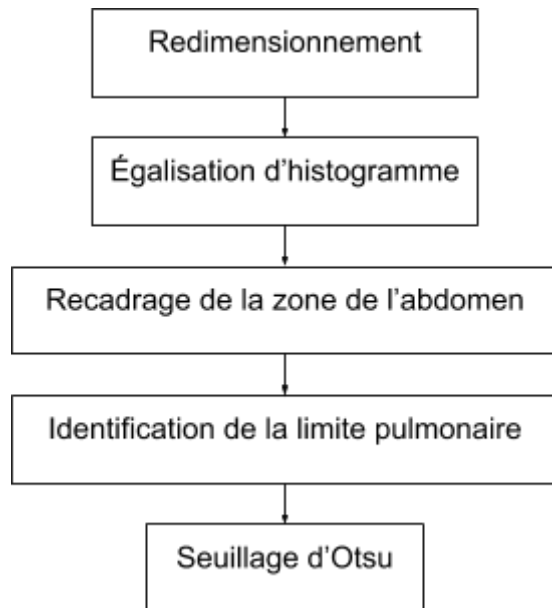


Sauf que des experts, il n'y en a pas forcément là où se trouvent les enfants atteints de cette maladie, c'est pourquoi nous aimerions développer un Système expert chargé de remplacer le professionnel de santé dans ce diagnostic.

Traitement de l'image et analyse de contours

Méthode

L'algorithme présenté dans cette partie est issu de la publication scientifique de 2017 : « [Detection of pneumonia clouds in chest X-ray using image processing approach](#) ». Il propose une approche du problème en 5 étapes :



Redimensionnement

Il a un double objectif, homogénéiser la taille des images et réduire le temps de traitement en réduisant la résolution

Egalisation d'histogramme

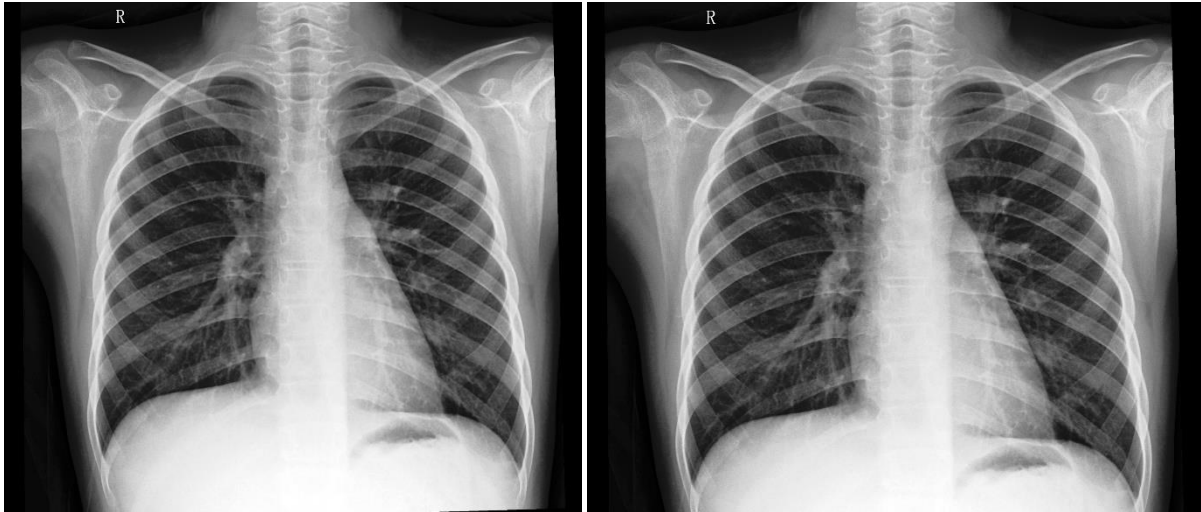
Ce traitement sert à faire ressortir les contrastes de l'image et ainsi à distinguer plus clairement les zones à analyser.



On remarque sur la deuxième figure que les zones correspondantes aux poumons sont plus contrastées que sur la première figure.

Recadrage

La région située sous les poumons et qui est entièrement blanche est coupée afin de retirer la région de l'abdomen, cette région n'est pas significative pour la détection de pneumonie.



Afin de trouver l'endroit à partir duquel on peut couper, on démarre du coin en bas à gauche et on parcourt horizontalement la radio par groupe de 5 pixels, on calcul l'intensité moyenne de chaque groupe de pixels au groupe précédent si le l'intensité diminue de plus de 50 alors on coupera sur cette ligne (début d'un poumon) si en bout de ligne cette condition n'est pas remplie on monte de 5 lignes et on recommence.

Identification

Afin d'identifier les limites pulmonaires, l'image est séparée horizontalement au milieu afin d'avoir un poumon sur chaque sous-image, puis chaque image est coupée en 100 lignes horizontales, sur ces lignes on refait une recherche de différence de gradient du blanc vers le noir supérieur a un certain nombre en reprenant 5 pixels pour limiter les faux positifs (minima/maxima local), les 100 points de chaque image permettent d'identifier le poumon plus ou moins précisément une fois reliés.

Exemple de point sur image gauche :



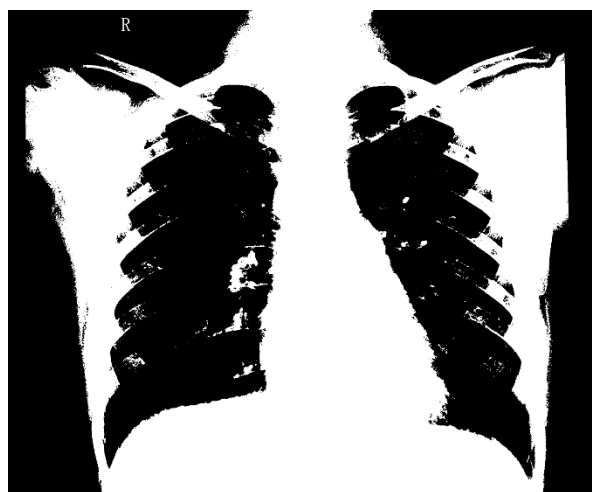
Cependant des problèmes de découpage peuvent se produire notamment si les bras sont apparents sur les côtés de l'image d'origine ainsi que parfois un suivi de côte :



Mais dans sa globalité l'algorithme fonctionne bien.

Seuillage d'Otsu

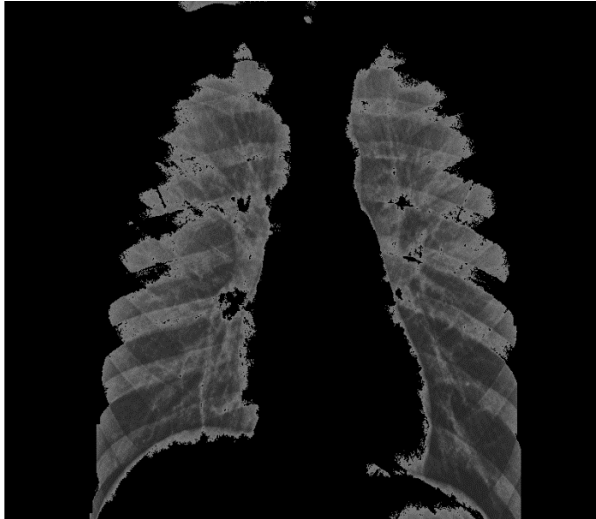
Le seuillage utilisé est celui d'Otsu. Il permet de séparer la partie saine du poumon de celle affectée par une pneumonie. Le seuillage d'Otsu aussi appelé Méthode d'Otsu sépare l'image en deux classes et cherche le seuil qui minimise la variance intra classe.



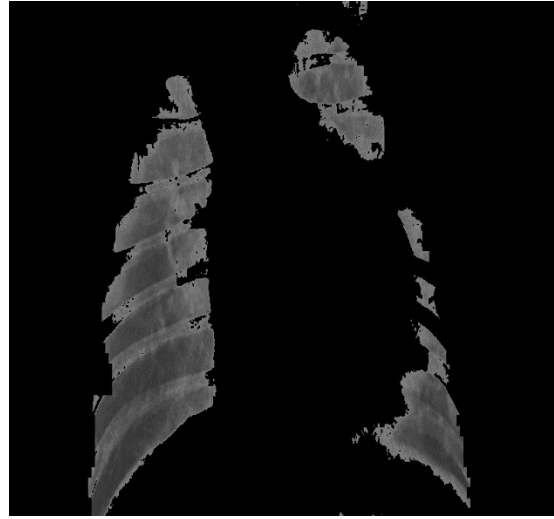
Combinaison et analyse

Afin de déterminer si un patient est malade, nous utilisons la surface des poumons récupérée lors de la partie 4, et cette surface après application du masque obtenu par le seuillage de l'étape 5 permet d'afficher la partie saine du poumon, un radiologue/médecin peut donc facilement voir si un patient est malade suivant la proportion de poumon affichée. L'algorithme prend le nombre de pixels du poumon avant application du masque et retire les nombres de pixels du masque afin d'avoir le nombre de pixels disparus. Si moins d'un certain pourcentage de l'image initiale a disparu alors le patient est sain sinon il est malade.

Personne saine



Personne malade



Limites et améliorations

Les résultats peuvent être faussés notamment à cause de l'étape de délimitation de la zone pulmonaire. En effet, sur certaines lignes, l'algorithme ne trouve pas la différence recherchée, suivant les radios des éléments extérieurs tels que les bras ou des appareils médicaux peuvent fausser la détection. De plus la détection se fait uniquement horizontalement sur toute la hauteur de l'image, ainsi est inclus en plus de la zone pulmonaire la partie haute du torse et le cou.

Résultats

On peut donner les résultats sur l'échantillon de test de façon suivante :

Prédit	Malade	26.93% (63 enfants)	67.69% (264 enfants)
	Normal	73.07% (171 enfants)	32.31% (126 enfants)
		Normal	Malade
Réel			

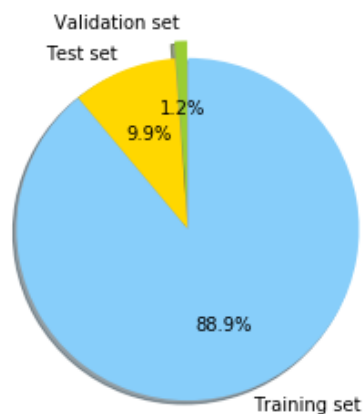
Réseau de neurones simple

Méthode

L'objectif de cette partie est de mettre en place un réseau de neurones ayant un bon ratio résultats/complexité. N'ayant pas des ressources de calculs énormes, nous avons préféré mettre en place une version « éducative », plus simple à entraîner.

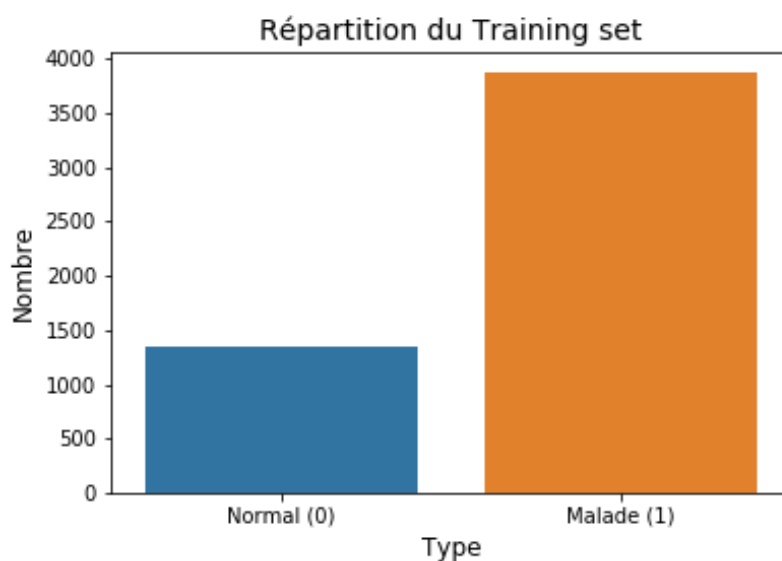
Analyse des données

Avant de passer à l'apprentissage, quelques vérifications de routine sur la base de données sont nécessaires. Tout d'abord la répartition dans les trois sets : Validation – Test – Train



Cette répartition n'est pas trop mauvaise et nous permet de garder un bon équilibre test/training.

Maintenant intéressons-nous plus en détails au training set et sa répartition des classes :



Cette présence non équilibrée des deux classes n'est pas vraiment étonnante, surtout dans le domaine médical, où l'on garde rarement les données des patients sains.

Cependant c'est pour nous un problème, qui risque d'entraîner un paradoxe d'acuité de la part du réseau lors de son entraînement.

Data-Augmentation

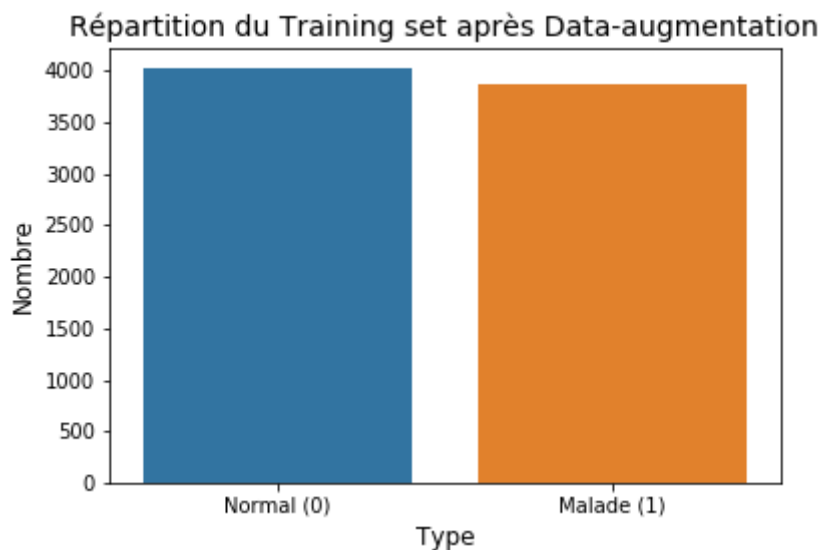
Pour pallier à ce problème de répartition nous allons utiliser un procédé dit de « Data augmentation » en créant de nouvelles images de patients sains. Pour cela on va utiliser des images de patients sains existantes et effectuer un traitement d'image mineur dessus.

Nous avons fait le choix d'exploiter aléatoirement 3 transformations mathématique :

- Rotation (De 20°)
- Symétrie verticale
- Variation des contrastes

Bien que cette solution ne soit parfaite, elle constitue un moyen artificiel permettant d'améliorer les résultats du réseau de façon non-négligeable.

Suite à la Data-Augmentation on revérifie la distribution des classes :



On a donc maintenant des données de qualité pour entraîner notre réseau.

Pré-traitement

On va ensuite rapidement effectuer quelques opérations habituelles :

- Standardisation en 150*150*3 des images
- Normalisation des images : Visant à accélérer la convergence du réseau

Nos images sont maintenant fin-prêtes à nourrir le réseau.

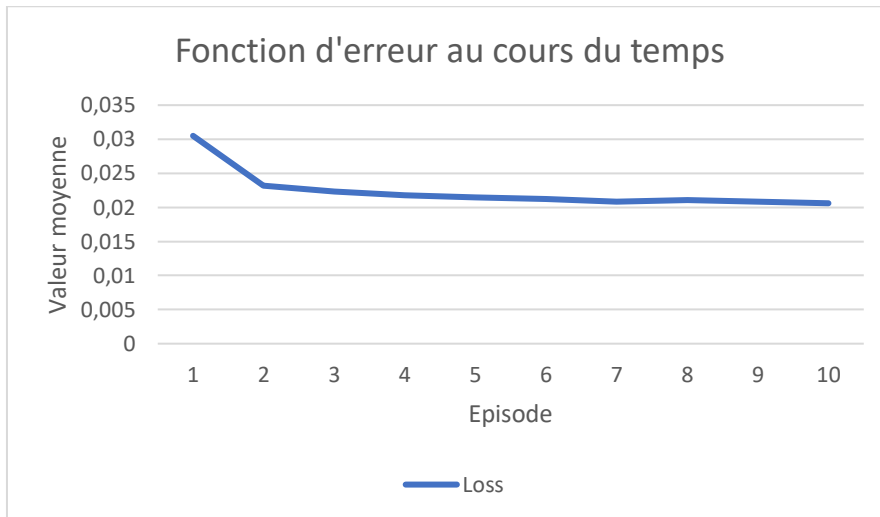
Architecture du réseau

Après de multiples essais, nous avons finalement choisis le modèle de réseau que vous pouvez retrouver dans le fichier « training.py ».

Entraînement

Concernant l'entraînement, nous utilisons les paramètres suivants :

- Taille de batch : 16
- Episode : 10
- Optimiseur : Adam
- Taux d'apprentissage : 0.0001
- Fonction d'optimisation : Entropie croisée



On observe bien un phénomène de convergence pendant l'entraînement.

Résultats

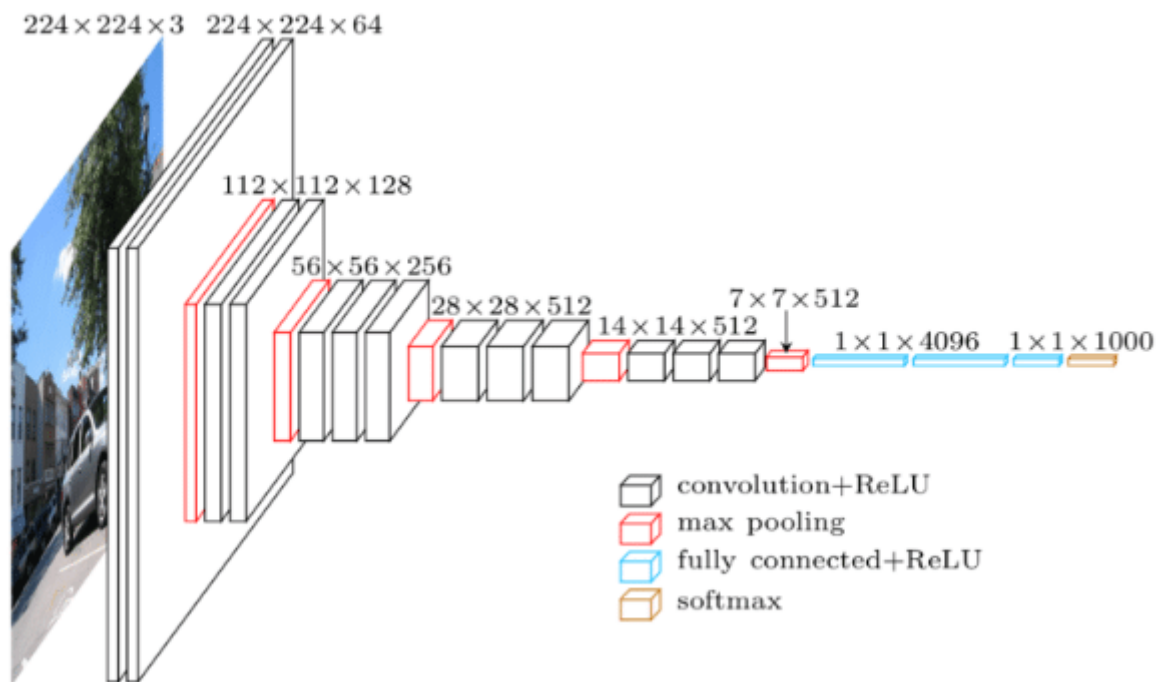
On peut donner les résultats sur l'échantillon de test de façon suivante :

Prédit	Malade	38.47% (90 enfants)	98.47% (384 enfants)
	Normal	61.53% (144 enfants)	1.53% (6 enfants)
		Normal	Malade
		Réal	

Ces résultats ne sont pas encore parfaits, mais ils sont déjà bien meilleurs que ceux obtenus précédemment.

Architecture poussée

Pour obtenir de meilleurs résultats, il nous aurait fallu utiliser une architecture plus lourde, en s'inspirant par exemple du réseau connu « VGG16 »



On aurait pu ainsi effectuer un transfert d'apprentissage des premières couches depuis une version pré-entraînée sur la base ImageNet car les premières couches servent principalement à la détection de motifs et de contours sur l'image.

On aurait également pu utiliser des couches de convolution séparables afin de limiter le nombre de paramètres à optimiser (Etant donné la faible taille de la base d'apprentissage).

Conclusion

Le projet à travers ces trois parties nous a permis de couvrir et de mettre en application une grande partie des connaissances acquises dans les deux modules (IN52 et IN54). Il nous a également permis de nous confronter à des problèmes pratiques d'implémentation et de recherche/optimisation de nos résultats.