



chpc

centre hospitalier public du cotentin



GREYC 

Projet MoNoMAD

Travaux de Natacha LAMBERT

Responsable CHPC : Dr. Laurette JITARU

Responsable GREYC : Abderrahim ELMOATAZ

Encadrant GREYC : Youssef CHAHIR

**Rapport sur les travaux de recherches effectués du 19 Mars 2018
au 18 Mars 2019 en tant que Ingénieure de Recherche hospitalier**



Introduction

Depuis plus de vingt ans, le laboratoire GREYC collabore avec l'équipe médicale du CHPC afin de pouvoir détecter les cellules anormales. En effet, avec l'avènement de l'informatique et d'outils puissants pour scanner des cellules microscopique à partir de prélèvements, il s'est avéré intéressant d'essayer d'automatiser le parcours d'une lame à travers la lunette d'un microscope afin d'y repérer les cellules susceptibles d'être cancéreuse. Celles-ci, peu nombreuse (moins d'1% par rapport aux dizaines de milliers de cellules présentes), sont souvent laborieuses à repérer. De fait, un traitement automatique pour cibler les endroits à vérifier serait un gain de temps en plus d'une aide précieuse pour éviter les faux négatifs.

C'est pourquoi cette collaboration, entre informaticiens et médecins, a vu le jour. Si l'approche a toujours été de segmenter l'image, soit trouver et détourner chaque cellule, pour ensuite la classer selon son type par réseau de neurones, l'approche ici est de ne plus utiliser la segmentation, trop fragile et chronophage, mais de se tourner vers les algorithmes d'apprentissage par deep learning.

C'est ainsi que le projet MoNoMAD naît, pour la période entre 2018 et 2020. Pour ce rapport, nous nous concentrons sur le travail de Natacha Lambert, embauchée pour ce projet du 19 Mars 2018 au 18 Mars 2019. Ce document relate les travaux effectués durant cette période et leurs résultats.

Nous commencerons par parler du contexte du projet, et le choix des technologies utilisées. Ensuite, nous verrons comment, grâce au deep learning, nous pouvons reconnaître des cellules (principalement leur noyau) par rapport à des débris, du cytoplasme ou des choses inexploitable. Cela comprend d'abord la fabrication d'une base de données assez conséquente d'images de noyaux et autres, à partir d'une lame de microscope comprenant toutes sortes de cellules, puis l'algorithme en lui même et enfin ses résultats. Nous verrons ensuite la détection de cellules sur une lame et ses résultats. Enfin, nous terminerons sur la reconnaissance des types de chaque cellule, tout d'abord entre Anormale et Normale, et ensuite selon des types plus précis avant de parler des améliorations et corrections effectués afin d'obtenir le meilleur résultat possible à l'heure actuelle. Nous concluerons enfin sur l'utilité de ces recherches concrètement, ainsi que sur les pistes à explorer afin de parfaire les résultats de cette année de travail.

Sommaire

Introduction	1
Contexte et principes du Deep Learning	3
Contexte	3
Principe du deep learning	4
Reconnaissance des cellules sur une lame	5
Préparation et fabrication de la Base de données	5
Algorithme utilisé	5
Description de chaque étape	6
Résultat et organisation	8
Apprentissage par deep learning	10
Utilisation de Tensorflow	10
Utilisation de Keras	10
Résultats	11
Détection de cellules sur une tuile	14
Détection grâce aux résultats précédent	15
Détection de germes directement depuis la tuile	17
Méthode avec Tensorflow	17
Méthode avec Keras	20
Résultats	21
Reconnaissance des types de cellules par le Deep Learning	24
Base de données fabriquée	24
Résultats	26
Reconnaissance multi-types de cellules	31
Reconnaissance générale des types	31
Reconnaissance des types des cellules normales	36
Augmentation de la base de données par rotation	40
Détection des types de cellules sur une tuile	44
Reconnaissance des amas par rapport aux cellules isolées	47
Base de données	47
Résultat	48
Réalité terrain de la reconnaissance de cellules anormales sur une tuile	50
Résultats	51
Corrections apportées	54
Pistes à explorer	57
Conclusion	59

Contexte et principes du Deep Learning

Pour commencer, nous allons voir qu'est ce que le deep learning, et pourquoi il est intéressant, dans notre situation, d'étudier cette méthode.

Contexte

Beaucoup de travaux ont été effectués depuis le début du projet de reconnaissance automatique de cellules anormales pour aider les pathologistes. Durant plus de vingt ans, différents projets et thèses se sont succédés afin de trouver les meilleurs solutions pour les médecins afin d'aider au diagnostic. De la collaboration entre le GREYC, l'Université de Caen, du CHU, du CHPC et de l'entreprise Datexim sortit plusieurs logiciels (Antartic, Cytoprocessor, Cytometror, ...) permettant de détecter et trier les cellules par type sur toute une lame, de les comparer et de poser ainsi un diagnostic. Aussi, d'autres outils (CellIA par exemple) permettent de créer facilement des bases de données et proposent des segmentations automatiques de cellules et d'y poser un classement correctible.

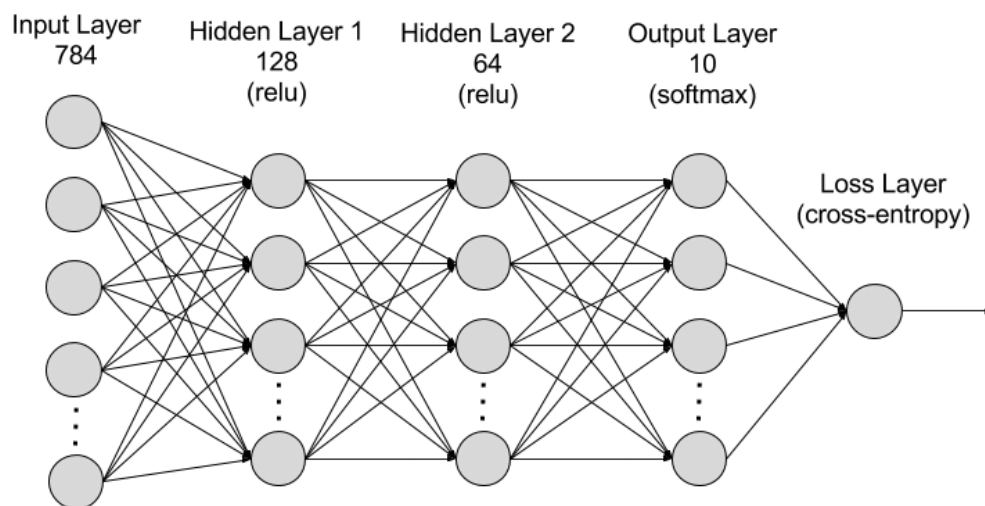
Cependant, tous ces projets visent à segmenter les cellules, soit détourner les noyaux, point intéressant à regarder pour poser le diagnostic. Or, il s'agit souvent du point faible des logiciels, puisque malgré des bons résultats et plusieurs algorithmes, ce processus est toujours long et dépend de beaucoup de paramètres. Beaucoup de recherches sont encore effectuées sur ce sujet, vu que l'oeil humain est bien plus performant que tout ce qu'on peut faire avec un ordinateur sur ce point.

Le but du projet MoNoMAD est donc de ne pas utiliser la segmentation et d'étudier s'il est possible de simplement reconnaître les cellules dans leur environnement, sans passer par le découpage de ladite cellule. Cette idée vient d'un algorithme qui permet de détecter les points sombres d'une tuile, donc des cellules. Si celui-ci est complété par une segmentation à l'heure actuelle, l'idée est de passer ce résultat par un algorithme de deep learning entraîné à reconnaître des cellules. En effet, la découpe en elle-même n'a pas d'intérêt puisque le but est de poser un diagnostic. De fait, si se passer de cette étape est possible, alors cela aurait beaucoup d'intérêt pour la détection de cellules en général, et pas seulement pour la gynécologie, principale domaine étudié par les précédents projets.

Le projet est donc d'exporter tous les travaux effectués auparavant pour tous les types de cellules, afin d'avoir une détection des cellules anormales quelque soit le cas, efficace et universel et ce par la technologie du deep learning.

Principe du deep learning

Le but premier de cette méthode est de reconnaître des objets comme le ferait le cerveau humain, c'est à dire en voyant suffisamment le même objet dans différents environnements pour en extraire les caractéristiques importantes. C'est pourquoi une grande base de données est requise, afin d'avoir le plus d'exemples possibles pour nourrir l'algorithme. A l'intérieur, la méthode essaie aussi d'imiter le cerveau humain en utilisant ce qu'on appelle un réseau de neurones : il s'agit d'une succession d'opérations mathématiques qui vont mettre en valeur certains points de l'image donnée. On parle de deep learning quand cette couche de neurones est assez profonde, donc avec beaucoup de neurones (qu'on appelle aussi convolution). Le schéma ci-dessous résume les couches successives :



A chaque passage d'une image, des caractéristiques vont être retenus. Plus celles-ci reviennent souvent, plus elles prennent de l'importance (par exemple, entre une pièce de Lego et un ballon, on peut retenir la forme rectangulaire ou ronde comme une caractéristique importante). Cependant, comme l'ordinateur apprend seul, nous n'avons pas la mainmise sur son apprentissage. L'humain lui fournit une base de données suffisante et annotée, afin que l'ordinateur sache ce qu'il apprend, et l'algorithme s'occupe du reste. A la fin de l'apprentissage, qui peut prendre du temps, l'ordinateur ressort un ou plusieurs fichiers condensant tout ce qu'il a pu apprendre.

Ensuite, nous pouvons passer à la classification. Grâce au modèle obtenu précédemment (les fichiers condensant les données apprises), l'algorithme va comparer une image donnée à ce qu'il connaît pour poser une décision sur ce que c'est, selon la probabilité que l'image appartienne à une classe ou non. Le bon fonctionnement de la classification dépend de l'apprentissage fait auparavant, sur une base suffisamment riche pour avoir le plus de cas possibles, et des caractéristiques bien discriminantes pour un classement fiable.

Dans notre cas, nous allons utiliser comme base d'apprentissage des imagerie contenant soit un noyau, soit des débris en tout genre. Nous avons donc deux classes différentes à apprendre à l'ordinateur. Nous verrons que nous allons utiliser 100 000 imagerie pour l'apprentissage, soit la moitié de notre base, et l'autre moitié servira de tests pour voir si l'apprentissage s'est bien passé. L'obtention d'autant de données ainsi que les méthodes précises utilisées seront décrites dans la partie qui suit.

Reconnaissance des cellules sur une lame

Maintenant que nous sommes familiers avec le Deep Learning, nous allons pouvoir à la fois créer une base d'apprentissage pour reconnaître les cellules puis apprendre sur celle-ci afin d'en évaluer l'efficacité.

Préparation et fabrication de la Base de données

Afin d'apprendre à l'ordinateur comment reconnaître une cellule du reste, il lui faut une base de données suffisamment conséquente. C'est pourquoi la première étape est de créer cette dite base, inexistante vu que les travaux précédents ne travaillaient pas avec l'apposition de germes, encore moins avec les objets qui ne sont pas des noyaux de cellules. Pour ce faire, la méthode semi-automatique a été choisie, dont les différentes briques vont être décrites dans la partie suivante. Ensuite, nous allons décrire comment chaque image a été rangée de manière pratique pour la suite des événements.

Algorithme utilisé

Tout d'abord, revenons sur la détection de germes, chose dont nous parlons depuis le début de cette partie. Il s'agit d'un travail effectué par Pierre Buysens, un ancien Ingénieur du GREYC dans l'équipe Image. Le principe est simple : il s'agit d'étudier les variations de couleurs de l'image pour détecter les pics colorés (soit les endroits les plus sombres) afin d'en sortir des points, qui sont les sommets de ces pics. Si cela est intéressant, c'est parce que ces pics correspondent généralement à des noyaux de cellules, base de nos recherches. A l'heure actuelle, l'algorithme est couplé avec les travaux sur la segmentation faits auparavant, en propageant le point trouvé afin de déterminer les bords du noyau.

Le point faible de cet algorithme est que les points posés, appelés germes, ne garantissent pas de se poser sur une cellule. En effet, l'algorithme est dépendant d'un paramètre, qu'un logiciel permet de faire varier pour les pathologistes. Selon sa valeur, des cellules peuvent être oubliées ou, au contraire, beaucoup de germes peuvent être posés à des endroits vides de cellules. C'est pourquoi notre première étape est de séparer cellules et non cellules : les germes posés seront ainsi triés par le classement que nous allons faire afin d'en retenir que les cellules.

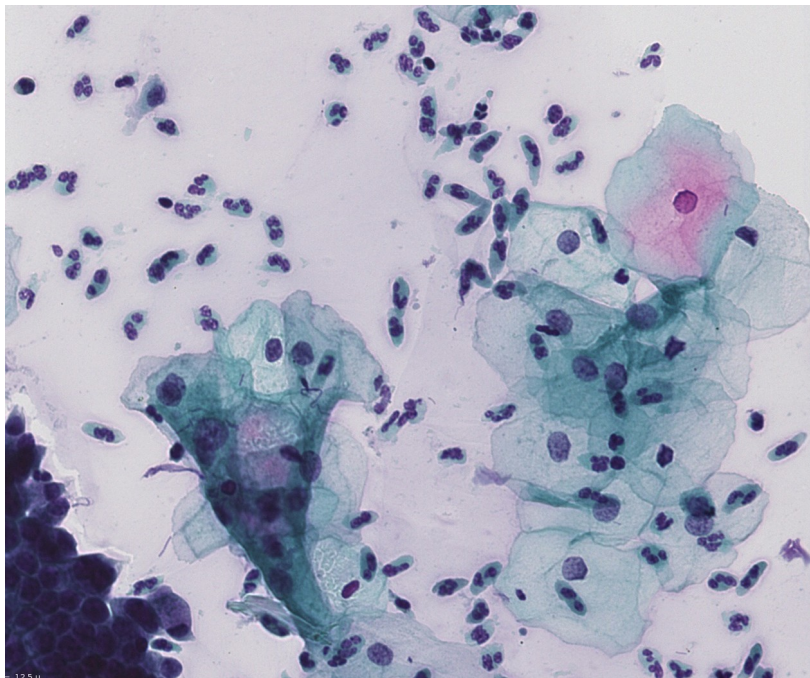
A partir de là, la marche à suivre est assez simple :

- Apposer des germes avec le paramètre au maximum (pour ne pas oublier de cellules)
- Découper des petites imagettes autour des germes (50 pixels sur 50 pixels)
- Sauvegarder ces imagettes pour l'apprentissage

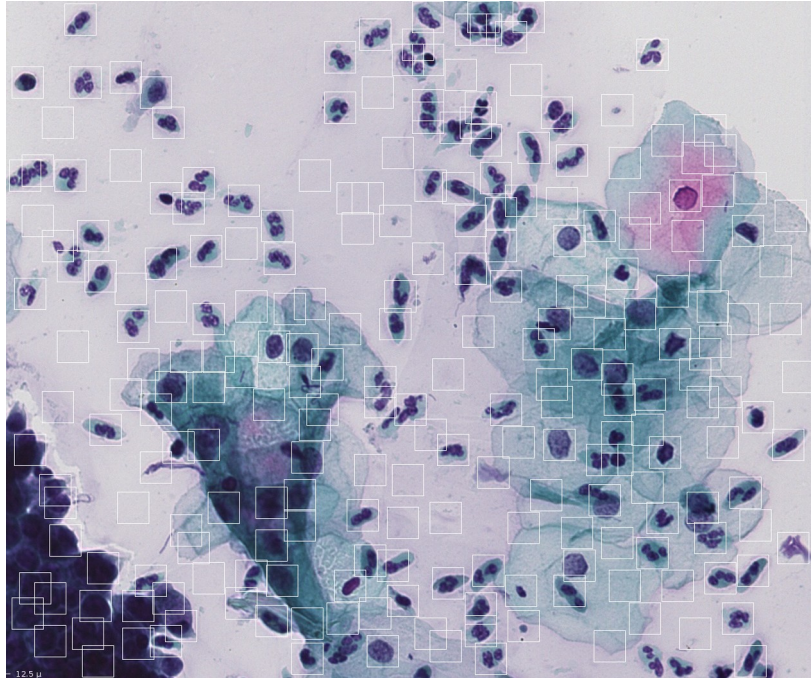
Pour notre base de données, il faut que ces imagettes soient annotées (soit indiquer s'il s'agit d'une cellule ou non). Il faut donc combiner les germes à une intervention humaine, soit la nôtre, puisqu'il n'y a pas besoin d'un expert pour reconnaître un noyau, afin d'annoter les germes qui possèdent une cellule par rapport à ceux qui n'en ont pas. Chaque imagette est sauvegardé ensuite selon ce résultat.

Description de chaque étape

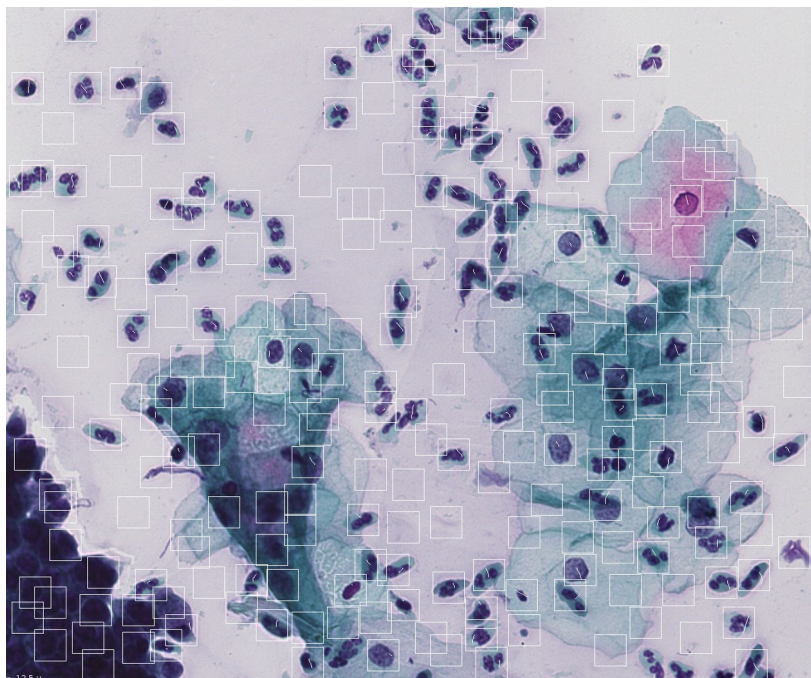
Prenons une image de cellules. Nous prendrons la tuile suivante, issue d'une lame de microscope pour illustrer la démarche suivie :



Sur cette image, nous allons appliquer l'algorithme de Pierre, qui va apposer des germes. Il s'agit d'un point blanc, de la taille d'un pixel, sur le milieu du pic trouvé. Pour plus de visibilité et afin de visualiser l'imagerie de sortie, nous allons afficher un carré autour du germe, prêt à être annoté. Nous obtenons donc cette image :

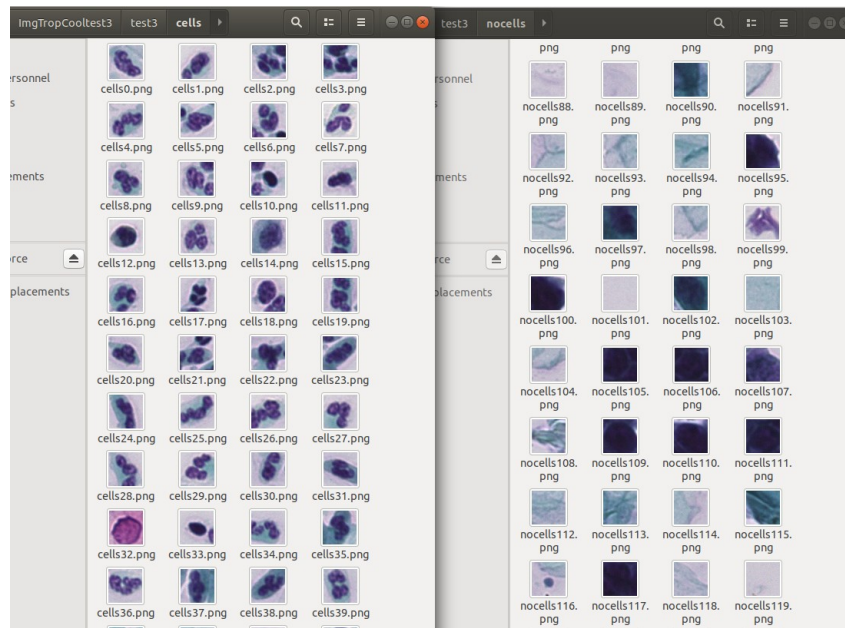


Comme nous pouvons le voir, les cellules sont bien encadrées, prouvant que les germes les ont trouvées, mais beaucoup de fond et de cytoplasme ont aussi été encadrés aussi. C'est pourquoi, à cette étape, il est requis une annotation humaine qui consiste à dessiner un point ou un trait dans les carrés qui correspondent à des cellules. A la fin de cette étape manuelle, nous avons cette image :



Nous pouvons valider cette annotation une fois sûr de n'avoir rien oublié. Il est à noter que tout ce qui est trop sombre n'est pas à prendre en compte, si rien n'est exploitable. En effet, si un pathologiste ne peut rien en faire, alors il n'y a aucune raison que l'ordinateur y arrive. De fait, il est plus intelligent de ne pas encombrer les médecins avec ces images.

Une fois cela fini, l'algorithme va découper chacun de ces carrés en imagerie, le tout rangé dans deux dossiers différents : un dossier pour les cellules, et un pour les non cellules. Cela peut se voir sur l'image ci-dessous :



Ainsi, nous avons nos imageries rangées de manière semi-automatique : avec un découpage automatique mais une annotation manuelle. Il suffit ensuite de lancer la même procédure sur une lame, préalablement découpée en tuiles. Si cette étape est longue, parfois fastidieuse au vue du nombre de cellules et de germes allant parfois jusqu'à 400 sur une même image, elle permet d'avoir en moins de deux semaines une base d'environ 200 000 imageries, moitié des cellules, moitié des non cellules.

Nous indiquerons tout de même que quelques erreurs, dû au grand nombre de carrés sur une petite zone, ayant pour conséquence de classer quelques cellules dans les dossiers non cellules. Aussi, comme la taille des imageries est fixe, certaines cellules se retrouvent coupées si elles ne sont pas centrées ou trop grandes. Enfin, comme l'erreur est humaine, certaines choses inexploitable peuvent être classées en cellules et inversement. Cependant, les quantités sont moindres, ce qui fait que cela n'entachera pas l'apprentissage, et les erreurs seront corrigées par la suite avec un expert, afin d'avoir une base sans erreur pour la version finale du projet.

Résultat et organisation

Après avoir réalisé la procédure précédente sur la totalité des tuiles d'une lame, soit environ 6400, nous avons notre base de données contenant exactement 202 373 imageries. Cependant, il est important de ne pas se contenter de ranger toutes ces imageries dans un seul dossier de manière anarchique. C'est pourquoi une architecture de rangement a été pensée afin de ne pas oublier d'informations importantes.

Tout d'abord, le dossier contenant le résultat de toutes les tuiles porte le nom de la lame, afin de pouvoir retrouver d'où les cellules proviennent. Ensuite, chaque dossier possède le nom de la tuile d'où les imageries sont issues.

Il faut savoir que les tuiles ont, lors du découpage, un nom à base de deux numéros qui correspondent au parcours de la lame en longueur et en largeur. Le premier numéro (de 0 à 79) correspond à la colonne tandis que le second numéro (aussi de 0 à 79) fait référence à la ligne. Le 0_0 se trouve donc dans le coin en haut à gauche de l'image.

Par contre, les tuiles ne contenant aucun germe ne possèdent aucun dossier. Pour vérifier le passage dans ceux-ci, histoire d'être sûr de n'avoir rien oublié, et récapituler le nombre de germes, de cellules et de non cellules, un document texte indique chaque passage effectué dans chaque tuile de cette manière :

Tuile : 31_22 / Nombre de cellules : 18 / Nombre de non cellules : 13 / Nombre de Germes : 31

Tuile : 31_23 / Nombre de cellules : 11 / Nombre de non cellules : 13 / Nombre de Germes : 24

Tuile : 31_24 / Nombre de cellules : 4 / Nombre de non cellules : 4 / Nombre de Germes : 8

Enfin, dans chaque dossier de tuile, nous avons un dossier contenant les cellules et un autre contenant les non cellules, comme montré dans la section précédente. Pour retenir le placement de chaque imagerie, afin de les retrouver sur la tuile pour visualiser dans son environnement, un document texte indique les coordonnées de chaque germe. Quant au nom des images, le numéro est donné dans l'ordre de rencontre. Cette architecture est résumée ci-dessous :

```
NomDeLaLame
  DocumentTxt de passage sur les tuiles
  NomdeDossier1
  NomdeDossier2
  ...
  NomdeDossier6399
  NomdeDossier6400
    DocumentTxt de coordonnées
    cells
      cells0.png
      cells1.png
      ...
      cells34.png
    nocells
      nocells0.png
      nocells1.png
      ...
      nocells25.png
```

Ainsi, nous avons notre base de données pour le deep learning. Pour la suite, afin de respecter les bonnes pratiques, nous avons séparé les dossiers en deux : d'abord la première partie (du dossier 0_0 au dossier 39_79) servira à l'apprentissage, le milieu (du dossier 40_0 jusqu'au dossier 43_79) servira soit de tests lors de l'apprentissage au besoin, soit sera intégré à la fin de la base de donnée (du dossier 44_0 au dossier 79_79) pour tester la classification apprise auparavant.

Apprentissage par deep learning

Nous allons parler ici de deux algorithmes issus de deux boîtes à outils différentes : Tensorflow, par Google, et Keras. Ce dernier utilise le premier en fond mais ne fonctionne pas de la même manière. Nous décrirons plus loin dans le document pourquoi avoir essayé deux algorithmes différents.

Utilisation de Tensorflow

Pour le premier algorithme, utilisant Tensorflow, l'utilisation est simple : l'apprentissage prend deux listes d'images, une pour chaque classe, et apprend dessus. Ensuite, environ 300 fichiers sont créés, résumant ce que l'ordinateur a appris. Puis, pour la classification, il suffit de charger ces fichiers et de passer une image et l'algorithme donne le résultat. Des adaptations pour avoir la liste des images, ainsi qu'appliquer le classement sur plusieurs images ont dû être fait, ainsi que quelques changements pour faire des statistiques et évaluer les résultats.

Si ce premier algorithme obtenait de bons résultats sur une petite base de données (60% de reconnaissances pour les cellules et 80% de reconnaissance des non cellules) au départ, il n'y eut aucun moyen de retrouver de tels résultats, quelque soit la base de données. L'apprentissage avait un taux d'erreur de plus de 90%. Ce souci étant indépendant de nous (aucune modification n'a été faite depuis la version qui fonctionnait), nous nous sommes tournés vers une autre méthode, qui nous indiquerait si le souci venait des données en elle même ou bien des outils utilisés sur lequel nous n'avons pas de maîtrise.

Utilisation de Keras

Le second algorithme utilise la boîte à outil Keras, qui utilise Tensorflow en fond. Mais la manière de procéder reste différente, c'est pourquoi il est intéressant de l'essayer. Le principe reste globalement le même, sauf que l'apprentissage demande un peu plus d'attention.

De fait, il faut fournir une architecture de dossiers particulière pour que l'ordinateur puisse apprendre; celui-ci demande deux dossiers différents, contenant chacun les classes à apprendre (pour nous, cellules et non cellules). Dans le premier dossier, qui compose la base d'apprentissage, il faut retrouver toutes les images au même endroit, soit dans un gros dossier cellules, et un autre gros dossier non cellules. Le second dossier est plus modeste, et est composé d'un nombre correspondant à 10% de la base d'apprentissage (donc pour

100 000 imageries, il sera composé de 10 000 imageries). Ce dernier sert à la validation de ce que l'ordinateur apprend : en effet, c'est une étape cruciale qui était automatisé selon la liste d'image dans le premier algorithme. Afin de vérifier que l'ordinateur apprend correctement, des imageries sont utilisées pour tester, comme nous le faisons après, si l'ordinateur trouve bien la bonne classe. Ces tests sont faits tout au long de l'apprentissage afin d'en retenir que le meilleur. Une fois cela fait, l'algorithme condense tout ce qu'il a appris sur deux fichiers.

La classification fonctionne de la même manière, avec un chargement des fichiers précédents. Il est important de noter que l'apprentissage avec cette méthode peut aller de 30 min à 3h selon la taille de l'image et le nombre d'itérations, contre 8h pour la méthode précédente, en prenant en compte que les calculs communiquent avec un disque dur externe, ralentissant la procédure. Cette méthode compte déjà un gain de temps considérable, même s'il faut noter que l'apprentissage ne se fait qu'une fois. Pour la classification, nous passons de 1h20 à une durée entre 20 min à 1h pour le même nombre d'imageries, toujours selon la taille et le nombre d'itérations.

Résultats

Si le premier algorithme n'obtient pas de résultats concluants pour des raisons obscures (moins de 50% de reconnaissance, même avec l'apprentissage sur une grande base de données), le second avait, au premier lancement, un taux de 90% de positifs.

Pour améliorer les résultats, nous avons modifié les paramètres de tailles des images (entre 16, 32, 64 et 128 pixels) ainsi que le nombre d'epochs (soit le nombre d'itérations de l'apprentissage), afin de gagner à la fois en temps de calcul, mais aussi en qualité de résultats. Le tout est résumé sur le tableau suivant :

	Taille des images en pixel	Nombre d'Epochs	Pourcentage de cellules bien classés (P)	Pourcentage de cellules mal classés (FN)	Pourcentage de non cellules bien classés (P)	Pourcentage de non cellules mal classés (FP)	Total Positif (P)
Test N°1	64	25	83.6	16.4	96.3	3.7	90.8
Test N°2	64	25	92.7	7.3	95.3	4.7	94.2
Test N°3	64	50	84.2	15.8	95.5	4.5	90.6
Test N°4	64	100	32.9	67.1	99.0	1.0	70.5
Test N°5	64	10	97.7	2.3	93.9	6.1	95.5
Test N°6	64	20	88.5	11.5	96.2	3.8	92.9
Test N°7	64	15	88.2	11.8	96.5	3.5	93.0
Test N°8	32	25	87.2	12.8	96.1	3.9	92.2
Test N°9	16	25	98.8	1.2	92.8	7.2	95.4
Test N°10	16	10	99.0	1.0	92.0	8.0	95.0
Test N°11	128	25	14.5	85.5	99.7	0.3	62.9
Test N°12	32	10	94.0	6.0	95.3	4.7	94.8

Test N°13	64	5	92.8	7.2	95.1	4.9	94.1
Test N°14	32	5	95.2	4.8	94.9	5.1	95.0
Test N°15	16	5	99.5	0.5	90.6	9.4	94.4
Test N°16	16	8	99.8	0.2	87.4	12.6	92.8
Test N°17	32	8	94.4	5.6	95.0	5.0	94.8
Test N°18	64	8	96.5	3.5	93.2	6.8	94.6
Test N°19	128	5	90.1	9.9	95.7	4.3	93.3
Test N°20	64	1	96.6	3.4	93.0	7.0	94.5
Test N°21	32	1	97.6	2.4	93.1	6.9	94.0
Test N°22	16	1	88.7	11.3	95.2	4.8	92.4
Test N°23	16	3	99.8	0.2	88.4	11.6	93.3
Test N°24	32	3	95.5	4.5	94.3	5.7	94.8
Test N°25	64	3	97.5	2.5	92.8	7.2	94.8

Les lignes surlignées en violet correspondent aux meilleurs résultats de façon globale, soit au minimum 95%. Les résultats indiqués en vert sont des très bons résultats en eux-même, ceux en jaune des résultats vraiment moyens (moins de 90%) et en rouge des mauvais résultats (moins de 75%). Les non colorés ne sont pas à signaler (au dessus de 90%, jusqu'à 96% selon les cas).

Comme nous pouvons le voir, les résultats sont plus que correctes, allant jusqu'à 95,5% de positifs dans la reconnaissance des cellules et des non cellules. Il est, par ailleurs, à noter que vu que nous travaillons sur une base de données pas encore vérifiée par un expert pour des raisons pratiques, une petite marge d'erreur est à prévoir dans le classement des cellules. Ces résultats, obtenus avec un algorithme perfectible au niveau des convolutions, pour les spécifier davantage pour les cellules, sont plus qu'encourageants pour la suite.

Nous pouvons aussi remarquer que la reconnaissance de cellules et des non cellules sont dépendantes l'une de l'autre. En effet, lorsque l'une augmente en taux de reconnaissance, la seconde est moins performante. Il est important de trouver un bon équilibre entre les deux, bien que l'important soit d'abord de classer toutes les cellules comme il faut pour ne pas en oublier et éviter les faux négatifs. S'il y a un peu plus de faux positifs, même s'il faut les limiter pour ne pas encombrer les pathologistes, cela est un moindre mal par rapport à ne pas remarquer une cellule anormale.

Le choix de l'apprentissage préférera donc quelque chose qui valorise les cellules. Les choix avec les images de 16 pixels sur 16 pixels, avec un nombre suffisant d'epochs, seraient des bons candidats puisque très bons pour les cellules (toujours plus de 98%) tout en étant bien plus rapide en apprentissage et en classification. Cependant, en améliorant l'algorithme, il se peut que les résultats changent. C'est pourquoi nous en garderons aussi avec des images plus grandes, afin de trouver le meilleur compromis, lors des améliorations de l'algorithme. Pour la suite, nous en prendrons trois pour voir les temps de calculs selon la taille. Dans un second temps, lors des améliorations, nous verrons à n'en choisir qu'une version.

Afin de mieux évaluer l'efficacité de l'algorithme, nous allons aussi tester l'apprentissage sur d'autres lames, qui sont très différentes de celle de la base, bien qu'il s'agisse toujours de cellules gynécologiques. Si nous atteignons 95%, ces chiffres ont pu être revu.

En effet, la première comporte beaucoup de cellules isolées, et surtout beaucoup de cellules avec un cytoplasme très vert et un noyau parfois peu visible, tandis que la seconde comporte essentiellement des amas de cellules (n'ayant pas pour classe "Amas" mais sont des regroupements de cellules dû à une mauvaise préparation de lame). Les deux tableaux ci-dessous résument les différents tests avec plusieurs modèles d'apprentissages vus auparavant. A noter que l'on a pris ceux avec les meilleurs résultats :

GC-18-254264-1-0-1_2018-04-19_14_25_15	Taille des images en pixel	Nombre d'Epochs	Pourcentage de cellules bien classés (P)	Pourcentage de cellules mal classés (FN)	Pourcentage de non cellules bien classés (P)	Pourcentage de non cellules mal classés (FP)	Total Positif (P)
Test N°5	64	10	92.1	7.9	90.2	9.8	91.3
Test N°9	16	25	93.6	6.4	89.0	11.0	91.5
Test N°10	16	10	97.3	2.7	82.3	17.7	90.6
Test N°12	32	10	85.8	14.2	92.2	7.8	88.7
Test N°14	32	5	89.3	10.7	91.3	8.7	90.2
Test N°17	32	8	87.5	12.5	92.2	7.8	89.6
Test N°24	32	3	85.7	14.3	91.8	8.2	88.4
Test N°25	64	3	94.7	5.3	87.1	12.9	91.4

GC-18-257024-1-0-1_2018-04-10_14_35_47	Taille des images en pixel	Nombre d'Epochs	Pourcentage de cellules bien classés (P)	Pourcentage de cellules mal classés (FN)	Pourcentage de non cellules bien classés (P)	Pourcentage de non cellules mal classés (FP)	Total Positif (P)
Test N°5	64	10	76.8	23.2	86.5	13.5	82.5
Test N°9	16	25	78.6	21.4	92.4	7.6	86.8
Test N°10	16	10	84.8	15.2	79.6	20.4	81.7
Test N°12	32	10	59.3	40.7	92.8	7.2	79.1
Test N°14	32	5	68.2	31.8	92.1	7.9	82.3
Test N°17	32	8	67.0	33.0	92.5	7.5	82.1
Test N°24	32	3	68.0	32.0	92.8	7.2	82.7
Test N°25	64	3	81.6	18.4	85.9	14.1	84.1

Tous les tableaux sont disponible sur ce lien : <https://drive.google.com/open?id=1ikbZOnqED6nU5DtHHmOkHBzHsfOsYIGE3zO6rMfNvzQ>

Les codes couleurs sont les mêmes que lors des résultats précédents. Comme nous pouvons le voir, les résultats sont moins bons en général.

Pour le premier cas, il s'agit de logique : le cas des nombreuses cellules très vertes n'a pas été vu lors de l'apprentissage ce qui fait que le modèle a du mal à trouver qu'il s'agit de cellules à cause d'une trop grande différence avec ce qu'il a l'habitude de voir. Les taux restant au dessus de 90%, il n'y a pas de quoi s'inquiéter d'un plus faible résultat. Celui-ci pourra être amélioré en incorporant à la base suffisamment de cellules différentes pour parer tous les cas.

Quant au second cas, il est expliqué par les amas. En effet, il est difficile de tirer quelque chose des amas de cellules, puisque souvent trop sombres pour distinguer quelque chose. Il est donc normal que l'algorithme ait du mal à décider si un amas est exploitable ou non, puisque l'humain a les mêmes difficultés. Il faut aussi noter que les annotations des cellules et non cellules ont été faites par un non expert, qui n'a pas forcément la bonne décision à ce sujet, augmentant les risques d'erreurs de classification. Si le taux chute dans les 80%, il n'est pas tant inquiétant au vu des soucis évoqués. Il faudra cependant tester à nouveau ce cas de figure, avec un expert.

Ces résultats nous présentent des classifications d'images, ce qui signifie qu'il faut au préalable découper chaque tuile en images, afin d'effectuer sur ces images la classification. Jusque là, ces images étaient annotées à la main, pour faire une base de données. L'étape suivante est donc d'automatiser la classification : une tuile est donnée à l'algorithme et il en ressort la tuile avec des carrés autour des cellules qu'il a détecté. Nous allons donc voir comment utiliser les travaux précédents pour faire cette détection.

Détection de cellules sur une tuile

Si reconnaître une cellule sur une image, c'est bien, il faudrait maintenant pouvoir la reconnaître directement sur la tuile à laquelle elle appartient pour ensuite la reconnaître directement sur la lame, afin de pouvoir utiliser notre bel algorithme dans une situation réelle.

Détection grâce aux résultats précédent

Pour détecter directement sur une tuile, il faut tout d'abord choisir un modèle. Au vu des résultats précédents, la version N°9 de l'apprentissage semble être le plus stable en résultat sur les trois lames, c'est donc celle-ci que nous allons utiliser.

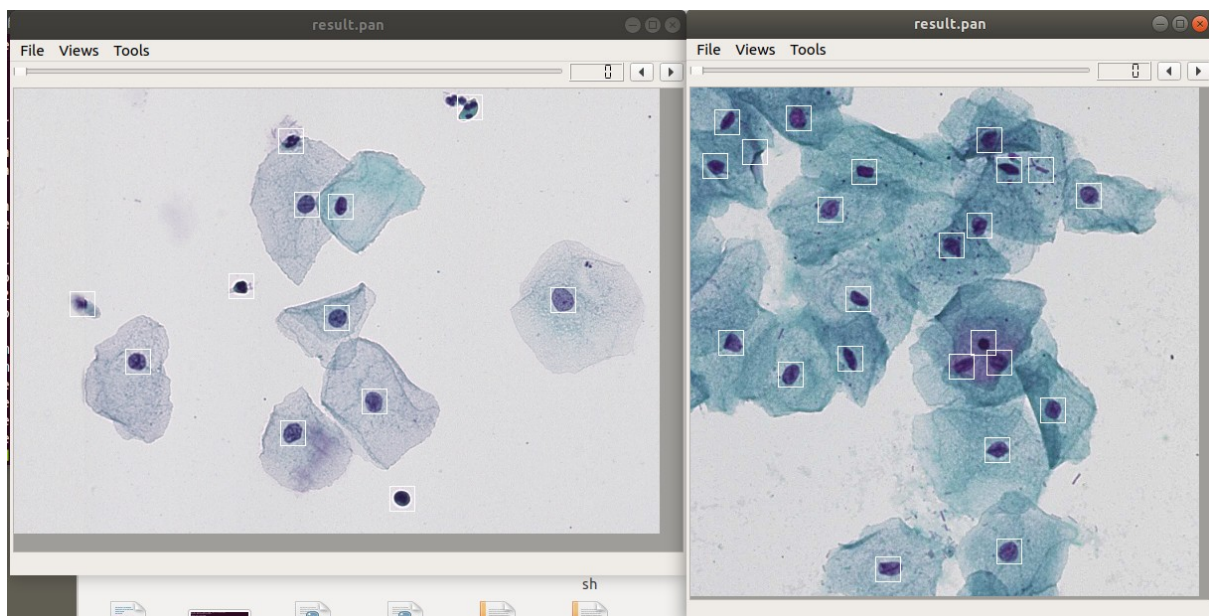
Ensuite, il faut savoir que la détection par deep learning fonctionne par découpage de l'image à traiter en différentes images, et chacune de ces images est passé dans l'algorithme afin de savoir si une cellule est présente ou non. Pour éviter de découper toute la tuile, en particulier du vide, présent au minimum de moitié sur une lame, nous allons faire

un découpage avec l'algorithme des germes, composé par Pierre, vu précédemment. Ainsi, la première étape sera de détecter les germes sur l'image donnée et d'en ressortir les imageries autour des germes. Il s'agit de ce qui a été fait précédemment, à la différence qu'il n'y a pas d'annotations, ce qui fait que les imageries sont rangées dans le même dossier, avec l'unique nom "cells" et leur numéro d'identifiant.

Par la suite, ces imageries sont passées au classifieur, qui va émettre un jugement sur s'il s'agit de cellules ou non. Pour afficher le résultat par la suite, l'algorithme prendra l'image de germes produite auparavant, afin d'effacer les germes correspondant à des non cellules. Ainsi, il ne restera que les germes correspondant aux cellules. Il sera aussi possible d'effacer les imageries ne contenant pas de cellules pour économiser de la place.

Enfin, de cette image de germe, nous affichons un carré autour, correspondant à l'imagerie. La détection de cellules se fait donc bien ainsi, du moins pour une tuile.

Afin de pallier à plus de cas, il sera aussi possible de sélectionner plusieurs tuiles d'une même lame pour les traiter simultanément. Elles s'afficheront ensemble à la fin du traitement. Il est à noter qu'elles doivent appartenir à la même lame, puisqu'elles seront rangées sous le même nom de lame. Il n'est, de toute façon, pas logique de traiter des tuiles de lames différentes. Le résultat donne ceci :



Enfin, le programme est prévu pour traiter toute une lame en même temps. Dans ce cas, l'affichage ne se fait pas et les résultats sont indiqués par écrit. En effet, lors de la recherche de germes et d'imageries, chaque germe selon sa tuile est notée dans un fichier .txt, avec ses coordonnées, ce qui permet de retrouver sa position exacte sur la lame. De même, chaque résultat de classification est noté dans un fichier au format CSV, comme effectué auparavant, avec les coordonnées du germe traité, toujours pour les mêmes raisons. De fait, le traitement sur toute une tuile visera d'abord à découper toutes les tuiles de la lame sous forme d'imageries, et de passer ces imageries dans le classifieur ensuite.

Cependant, cette étape est très longue (6h pour le découpage, 2H pour la classification) et peut planter. En effet, après plusieurs essais sur disque dur externe, celui-ci indique qu'il a terminé sans que cela soit vrai et bloque ainsi toute interaction avec. Le risque est de corrompre le disque dur externe en le retirant trop vite, ou bien de perdre une partie des données (en particulier les imageries), qui n'ont pas eu le temps de vraiment se sauvegarder. Le plus probable est qu'il s'agisse d'opérations non terminées, mises en parallèle. Par contre, ce problème peut être amélioré de différentes façon, notamment en parallélisant dans le code le processus, vu que les traitements peuvent facilement l'être, et en utilisant un disque local. Il restera en l'état pour le moment, mais fera le sujet d'améliorations plus tard.

Comme nous avons un fichier CSV contenant les résultats de la classification pour une tuile, et pas d'affichage pour une lame, il est totalement possible de n'afficher qu'une ou plusieurs tuiles, sans refaire le traitement. En effet, le script propose, à partir du CSV et de l'image, de recréer une image de germes pour affichage selon les résultats de la classification faits auparavant. De même que pour le traitement de plusieurs tuiles, il faut qu'elles soient traitées dans le même CSV, donc appartenir à la même lame.

Nous avons donc 5 scripts différents selon le nombre de tuiles à traiter. Le tout a été rassemblé dans un et même script, qui lancera, selon ses arguments, le bon script à appliquer. Si toutes les vérifications de bonne utilisation n'ont pas été faite, cela suffit pour un usage averti. Ainsi, nous possédons un algorithme capable de faire de la détection de cellules et d'afficher le résultat. Avant de passer à la détection des types de cellules, demandant une base de données conséquente, pas encore prête, nous allons nous diriger vers une autre branche : la détection des cellules mais sans les imageries.

Détection de germes directement depuis la tuile

Jusque là, nous avons utilisé un algorithme proposé par Pierre, qui va détecter les germes d'une image, soit les points foncés tel le charbon, comme nous avons expliqué dans la partie précédente. Le but ici est de se passer de cet algorithme, et d'utiliser le deep learning pour directement détecter les germes.

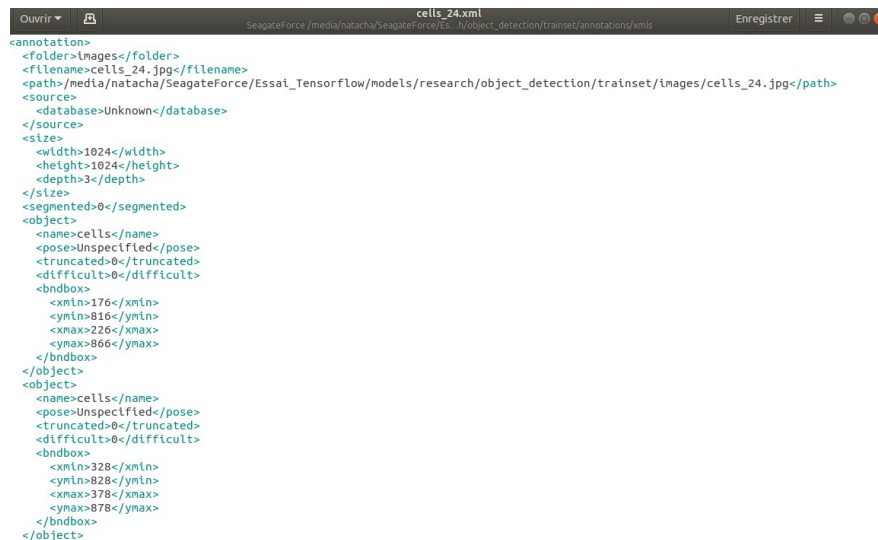
Méthode avec Tensorflow

Afin de tester et maîtriser la détection sur une image, nous allons d'abord travailler à détecter les cellules, et non pas les germes (soit une boîte englobante plus grande). Pour ne pas partir de rien et tester le fonctionnement d'un tel algorithme, nous suivrons le tutoriel disponible sur ce lien : <https://medium.com/@aescart1/la-computer-vision-d%C3%A9tection-dobjet-avec-tensorflow-5ee819d74f61> ainsi que Tensorflow.

La première étape est de constituer une base de donnée suffisante pour l'apprentissage. Dans notre cas, nous pouvons utiliser l'algorithme précédemment présenté de classification pour créer notre base de manière automatique. Il y aura un pourcentage d'erreur mais cela ne devrait pas poser de souci, vu qu'il est assez faible. Nous prenons une lame ressemblante à celle qui sert de base de données, afin d'être sûr d'atteindre un assez

bon pourcentage de bonne classification. Comme ce nouvel algorithme a besoin des images et d'un document XML contenant les boîtes englobantes des cellules, nous n'utiliserons que les résultats du CSV.

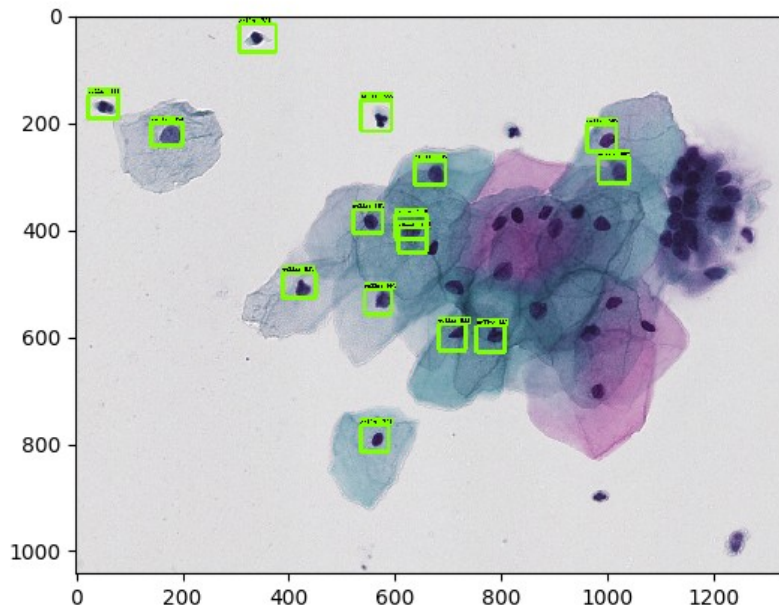
Ainsi, pour chaque tuile, une sauvegarde de ladite image est effectuée, au format JPG, ainsi que la liste des cellules détectées, avec une boîte de 50 pixels sur 50 pixels. Le document XML se présente sous cette forme :



```
<annotation>
  <folder>images</folder>
  <filename>cells_24.jpg</filename>
  <path>/media/natacha/SeagateForce/Essai_TensorFlow/models/research/object_detection/trainset/images/cells_24.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1024</width>
    <height>1024</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>cells</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficulty>0</difficulty>
    <bndbox>
      <xmin>176</xmin>
      <ymin>816</ymin>
      <xmax>226</xmax>
      <ymax>866</ymax>
    </bndbox>
  </object>
  <object>
    <name>cells</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficulty>0</difficulty>
    <bndbox>
      <xmin>328</xmin>
      <ymin>828</ymin>
      <xmax>378</xmax>
      <ymax>878</ymax>
    </bndbox>
  </object>
</annotation>
```

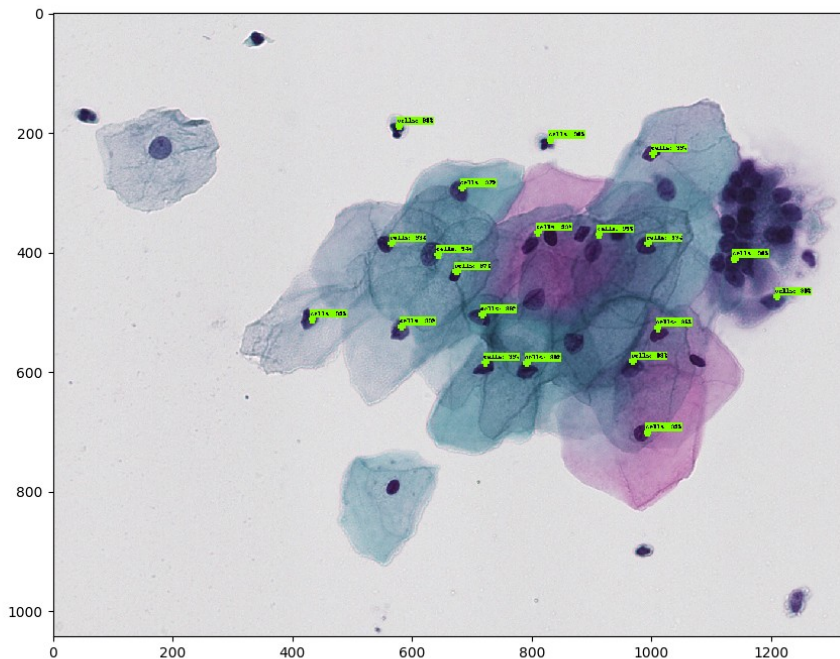
Une fois cela fait automatiquement, il est possible de lancer l'apprentissage. Il est à noter qu'il faut respecter une certaine architecture et définir certains documents .txt avec des valeurs de labels et de noms d'images. Mais ceci est bien décrit dans le tutoriel. Cette version d'apprentissage est, par contre, infini. Le but est d'étudier le résultat des pertes afin d'arrêter à l'étape qui nous convient. Dans notre cas, nous choisirons l'étape 2553. Ce choix est fait dû à la stabilité de l'algorithme entre 2 et 5 pendant des centaines d'étapes, signe que ce ne sera pas possible de faire mieux. Un script est prévu pour sauvegarder le modèle obtenu à cette étape.

Enfin, il suffit de lancer un script chargeant le modèle et l'appliquant sur une image. Cela affichera alors la zone contenant une cellule, avec le pourcentage de reconnaissance, comme nous pouvons le voir ci-dessous :



Le résultat nous montre qu'il est, malheureusement, bien moins efficace que les germes. En effet, nous pouvons voir des oublis de cellules simples, ainsi qu'une mauvaise détection des cellules proches. Le pourcentage montre aussi des difficultés sur des cas simples, mais difficile d'en savoir la cause. La taille des boîtes englobante pourrait en être une, par exemple, tout comme la forme patatoïde de certaines cellules.

Un deuxième apprentissage a été fait, en utilisant que les germes cette fois. Au lieu de mettre des boîtes englobantes autour des cellules, nous la mettons autour des germes, qui représentent le milieu d'une cellule. Nous avons donc des points de quelques pixels de dimensions à passer à l'algorithme. Pour plus de résultat, nous laissons tourner l'apprentissage trois fois plus longtemps, soit jusqu'à l'étape 6472. Malheureusement, nous sommes loin d'un résultat aussi satisfaisant qu'avec les imagerie, comme le montre cette image :



Comme Tensorflow n'a pas fonctionné lors des imageries, nous allons tenter une autre méthode, avec Keras, afin d'espérer de meilleurs résultats.

Méthode avec Keras

Comme pour la partie précédente, nous n'allons pas partir de rien. Nous allons donc prendre l'algorithme d'apprentissage utilisant Keras disponible sur ce dépôt : <https://github.com/fizyr/keras-retinanet> . De même, pour faire l'apprentissage, il faut de nouveau constituer une base de données, qui se présente sous la forme de deux CSV cette fois. Le modèle demandé est celui-ci :

CSV pour les annotations : path/to/image.jpg,x1,y1,x2,y2,class_name

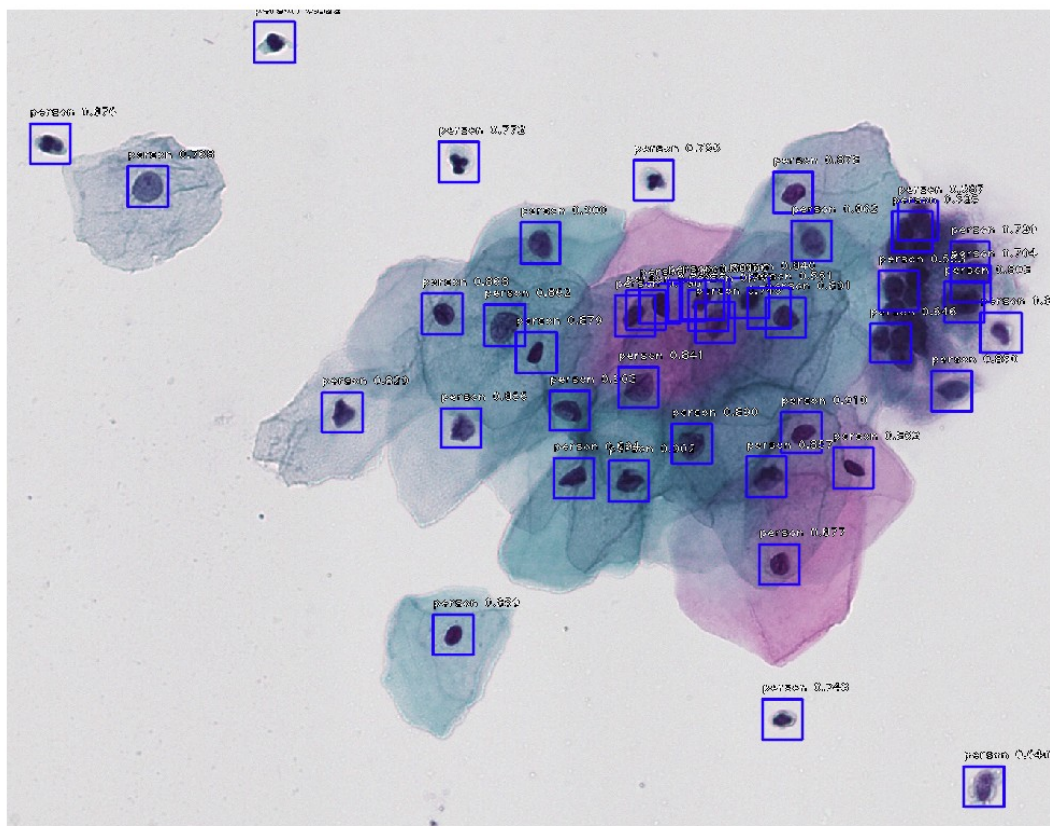
CSV pour la liste de classes : class_name,id

Le premier CSV doit comporter le chemin pour accéder à l'image et les coordonnées de la boîte englobante autour de l'objet à reconnaître et, pour finir, la classe à laquelle l'objet appartient. Si une image peut avoir plusieurs boîtes englobantes, il faut réaliser une ligne pour chaque objet. Dans notre cas, nous avons donc la même image répétée plusieurs fois, avec des coordonnées différentes pour les boîtes englobantes.

Le second CSV contient la liste des classes à apprendre. Dans notre cas, nous n'en apprenons qu'une : les cellules. Ainsi, il ne reste qu'à lancer l'apprentissage avec une commande. Cependant, par rapport aux autres calculs, le temps réservé à l'apprentissage

est très long : il faut environ 15 à 18 heures pour un apprentissage avec 25 Epochs et 250 steps. En comparaison, il faut un peu plus d'une heure pour la même chose à notre premier algorithme Keras sur les imagerie pour apprendre à différencier les non cellules des cellules. De même, le lancement d'un test est plus long que pour une imagerie. Cela est dû à la taille en partie, car nous travaillons avec des images de tuiles entières (soit 1024 pixels sur 1024) alors que les imagerie étaient au maximum à 64 pixels sur 64. Cependant, pour les prochains tests, les calculs seront répartis sur différentes machines du GREYC pour limiter le temps de traitement.

Si, d'après nos premiers tests avec un apprentissage trop court, nous n'obtenions aucune reconnaissance d'objets, que ce soit cellule, chien ou chaussette, un long calcul permet d'obtenir ce résultat, pour la même image de test que l'algorithme utilisant Tensorflow :



Comme nous pouvons le voir, toutes les cellules bien reconnaissables ont été repérées, contrairement aux essais précédents. S'il y a des erreurs, avec des prises en comptes de non cellules, ainsi que des cas douteux avec les amas, nous obtenons tout de même un résultat très satisfaisant visuellement.

Si ces résultats sont obtenus en comptant la boîte englobante autour des cellules, plusieurs tests ont été fait pour détecter le germe (soit quelques pixels au milieu de la cellule). Cependant, l'algorithme ne trouvait rien à moins de 20 pixels de rayon (soit le rayon

d'une petite cellule). La détection de germes paraît donc compromise, c'est pourquoi nous garderons la boîte englobante de la cellule plutôt que du germe afin de garantir un meilleur fonctionnement.

Résultats

Puisque les résultats avec Keras ont l'air visuellement intéressants, nous allons chiffrer la reconnaissance des cellules. Pour ce faire, nous allons comparer la reconnaissance de l'algorithme précédent sur les imagerie et celle-ci sur la même base de cellules avec les mêmes paramètres. Pour que le test soit complet, nous allons tester à la fois le nombre de cellules reconnues par rapport au nombre attendu, mais aussi comparer les positions trouvées et attendues, avec une marge d'erreur d'une dizaine de pixels (sachant qu'une cellule peut en faire 20 en minimum généralement).

Pour ce faire, le format CSV est utilisé. Nous aurons un CSV contenant la position des cellules attendues et leur nombre selon chaque tuile, et dans l'autre le résultat. Nous prendrons soin, avec une distance euclidienne, de comparer la position d'une cellule trouvée par l'algorithme avec celles attendues pour lui assigner la plus proche (si elle n'est pas trop éloignée). Ainsi, nous pourrons voir lesquelles n'ont pas été reconnues mais aussi celles qui ont été trouvées en trop.

Le tout a été compilé dans le tableau ci-dessous :

GC-15-234040-1-0-1_2015-09-25_15_41_56	Nombre d'étapes	Nombre d'Epochs	Cellules attendues	Cellules trouvées	Nombre de cellules attendues trouvées		Nombre de cellules attendues non trouvées		Nombre de cellules non attendues trouvées		Nombre de cellules "en trop" par rapport au nombre attendu	
Test N°1	250	25	93581	109938	79173	84,60%	14408	15,40%	30760	28%	16357	14,90%
Test N°5	250	10	93581	105407	77920	83,30%	15661	16,70%	27482	26,10%	11826	11,20%
Test N°9	1000	25	93581	101343	81046	86,60%	12540	13,40%	20297	20%	7762	7,60%
Test N°10	1000	10	93581	102895	81113	86,70%	12468	13,30%	21777	21,20%	9314	9,10%
Test N°14	500	5	93581	105619	78366	83,70%	15215	16,30%	27248	25,80%	12038	11,40%
Test Germes	250	10	93581	67252	62452	66,70%	31129	33,30%	4795	7,10%		

Nous avons utilisé les meilleurs paramètres pour la détection des cellules, vus dans la partie précédente. Tout d'abord, nous pouvons constater que ce nouvel algorithme trouve beaucoup plus de cellules qu'attendues, jusqu'à 15% en plus. Là où le précédent algorithme avait effectivement tendance à en prendre plus, mais éliminait les non cellules à 95%, nous avons beaucoup plus de choses erronées ici, puisque plus de 20% des cellules trouvées ne correspondent pas à des cellules. De même, seulement 85% des cellules attendues sont trouvées, soit bien moins que les 95% de l'algorithme précédent. Si les

résultats ne sont pas mauvais, ils sont bien moins efficaces qu'avec la précédente façon de faire, qui combine la détection de germes puis la détection sur ces imageries trouvées.

Des tests complémentaires ont été faits, comme sur la partie précédente, sur les autres lames :

GC-18-254264-1-0-1_2018-04-19_14_25_15	Nombre d'étapes	Nombre d'Epochs	Cellules attendues	Cellules trouvées	Nombre de cellules attendues trouvées		Nombre de cellules attendues non trouvées		Nombre de cellules non attendues trouvées		Nombre de cellules "en trop" par rapport au nombre attendu	
Test N°1	250	25	15731	19080	14270	90,70%	1461	9,30%	4810	25,20%	3349	17,60%
Test N°5	250	10	15731	19617	14428	91,70%	1303	8,30%	5189	26,50%	3886	19,80%
Test N°9	1000	25	15731	17013	14112	89,70%	1619	10,30%	2901	17,10%	1282	7,50%
Test N°10	1000	10	15731	17683	14303	90,90%	1428	9,10%	3380	19,10%	1952	11,00%
Test N°14	500	5	15731	18630	14121	89,80%	1610	10,20%	4509	24,20%	2899	15,60%
Test Germes	250	10	15731	13650	12298	78,20%	3433	21,80%	1352	9,90%		

GC-18-257024-1-0-1_2018-04-10_14_35_47	Nombre d'étapes	Nombre d'Epochs	Cellules attendues	Cellules trouvées	Nombre de cellules attendues trouvées		Nombre de cellules attendues non trouvées		Nombre de cellules non attendues trouvées		Nombre de cellules "en trop" par rapport au nombre attendu	
Test N°1	250	25	10591	10886	7302	68,90%	3289	31,10%	3584	32,90%	295	2,70%
Test N°5	250	10	10591	11152	7469	70,50%	3122	29,50%	3683	33,00%	561	5,00%
Test N°9	1000	25	10591	8981	7073	66,80%	3518	33,20%	1908	21,20%		
Test N°10	1000	10	10591	9676	7432	70,20%	3159	29,80%	2244	23,20%		
Test N°14	500	5	10591	9407	6784	64,10%	3807	35,90%	2623	27,90%		
Test Germes	250	10	10591	6107	5445	51,40%	5146	48,60%	662	10,80%		

Si pour l'un des cas, nous avons de meilleurs résultats, cela reste toujours inférieur à ce qu'on avait précédemment. Le second cas, par contre, est bien moins efficace et ne trouve même pas suffisamment de cellules. Cela nous montre que si la détection de cette manière était prometteuse, elle n'est cependant pas suffisante pour battre les très bons résultats précédents.

En ce qui concerne le temps de calcul, cette méthode n'est pas plus rapide à effectuer. En effet, l'apprentissage est bien plus long, même si cela n'est pas un problème en soi, mais l'algorithme n'est pas non plus plus efficace pour une lame. Nous pouvons compter aussi 8 heures pour traiter une lame, comme la méthode précédente. Enfin, cette méthode a tendance à dédoubler les cellules, là où la méthode précédente était bien plus

robuste à ce sujet, même lors des tuiles contenant jusqu'à 300 cellules polynucléaires. Quelques cas montrent aussi que certaines tuiles oublient jusqu'à un quart des cellules présentes dans ce genre de tuiles, sans aucune raison apparente.

Il est à noter que le test sur les germes a été ajouté au tableau. Nous pouvons voir qu'il manque beaucoup de cellules à la détection en utilisant juste le germe, alors que celui-ci a la taille d'une petite cellule. Vu que l'algorithme ne fonctionnait pas sans une taille suffisamment conséquente, le résultat est logique puisqu'avec cet apprentissage, les grosses cellules se seront pas reconnues. De plus, par rapport à la détection de germes par Pierre, cette méthode est plus longue.

En conclusion, rien dans cette méthode n'est supérieur à ce qui a été trouvé précédemment. Nous pouvons donc bannir cette technique et garder ce qui a été fait auparavant.

Reconnaissance des types de cellules par le Deep Learning

Une autre grande étape du projet consiste à pouvoir reconnaître les types des cellules, afin d'en retirer les anormales. Si quelques tests provisoires ont pu mettre en valeur pertinence et le potentiel de cette idée, nous allons approfondir cette voie avec une base de données un peu plus conséquente.

Base de données fabriquée

Pour pouvoir lancer un apprentissage sur les types de cellules, il faut d'abord avoir une base de données. Celle-ci est plus compliquée à obtenir puisqu'il est nécessaire qu'un expert construise cette base. Cependant, le manque d'outils efficace a ralenti cette fabrication, ce qui a pour effet de devoir manipuler les objets cellulaires afin d'avoir suffisamment de données pour nourrir l'algorithme.

Pour ce faire, nous allons utiliser deux bases. La première est la base utilisée par Datexim il y a quelques temps, réalisé par le Docteur Hubert Elie. Elle comporte des grandes images avec, au centre, la cellule visée, ainsi que son type et son anomalie. Pour plus de sûreté, cette base a été re-vérifiée par Michel Lécluse, notamment pour y apposer les types (autre que Normale et Anormale). Nous avons donc, grâce à cela, 1014 cellules anormales et 3783 cellules normales.

Des tests préliminaires ont été effectués sur cette base. Comme il s'agit de redondance sur ce qui va suivre, tout le détail ne sera pas décrit ici mais vous pouvez retrouver les résultats provisoires sur le lien suivant : https://docs.google.com/spreadsheets/d/1NAC6GzY21hu8I6G3OYreJBK52eAwBwmaCxFOsg_jL3I/edit?usp=sharing

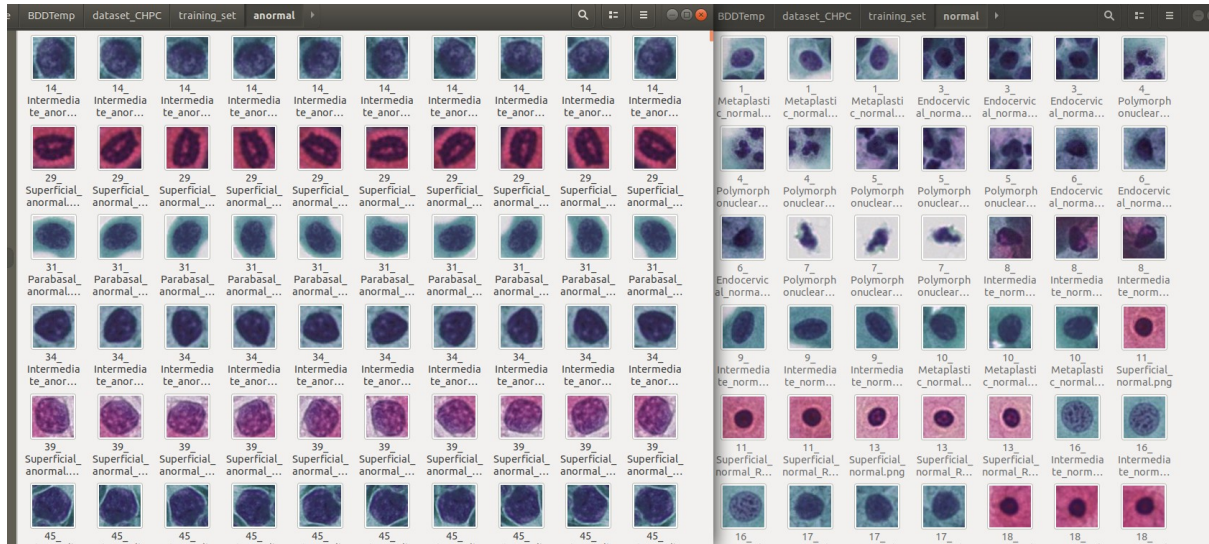
La seconde base est issue de CellIA, un outil développé pour faire rapidement et facilement des bases auprès d'un expert cytologique. Celle-ci est en cours de fabrication et sera mise à jour régulièrement selon le travail de Michel Lécluse. Elle différencie tous les types de cellules, bien que nous n'utiliserons pas cette donnée maintenant, sauf statistiquement. Elle comporte 428 cellules anormales et 1279 cellules normales.

Vu que cela ne sera pas suffisant pour avoir à la fois une base d'apprentissage et de tests, des rotations sur les cellules vont être effectuées pour agrandir la base de données afin d'avoir, dans chaque catégorie, environ 14 000 cellules. Afin d'équilibrer la base avec différents types de cellules (notamment pour les 6 types de normales), nous allons effectuer un nombre de rotations différent selon leur nombre de base. Ceci fait, ainsi que la répartition par type, sont décrit dans le tableau ci-dessous :

Base		Base CellIA	Base CytoPro	Total	Coef	Nombre total pour la base
ANORMAL		428	1014	1442	10	14420
	ASCUS	33	-	-		330
	HSIL	307	-	-		3070
	LSIL	88	-	-		880
NORMAL		1279	3783	5062		14678
	Dystrophies	55	381	436	6	2616
	Endocervicale	29	640	669	4	2676
	Intermédiaire	386	1089	1475	2	2950
	Parabasale	24	160	184	10	1840
	Polynucléaire	405	733	1138	2	2276
	Superficielle	380	780	1160	2	2320
TOTAL		1707	4797	6504		

Comme nous prenons en compte que les imagerie, chaque image donnée sera donc redécoupée, en plus de subir des rotations, afin de ne prendre en compte que la cellule concernée. La taille minimale sera de 64 pixels sur 64, mais environ 10% des cellules anormales auront des boîtes plus grandes, allant jusqu'à plus de 100 pixels d'envergure, pour les avoir en entier. Cependant, à l'apprentissage, les tailles seront remises à 64, 32 ou 16 pixels selon le choix que nous ferons.

Nous obtenons donc la liste des imagerie suivantes :



Chaque type de cellule sera indiqué dans son nom, ainsi que sa normalité et sa provenance entre les deux bases afin de pouvoir la tracer facilement grâce à son identifiant. Cela a été fait aussi pour simplifier les tests de l'apprentissage d'une part, et d'autres parts car les numéros d'identifiants entre les deux bases sont communs.

Il ne manque qu'à lancer le même algorithme que la reconnaissance de cellules. Pour nos tests, nous fonctionnerons de manière binaire, soit de savoir si une cellule est anormale ou non. A l'avenir, nous utiliserons plus de classes afin de reconnaître plus précisément les types.

Nous divisons cette base de données en deux : environ 7 000 images serviront de base d'apprentissage tandis que le reste servira pour les tests. Chaque cellule qui a subi une rotation est gardée dans sa famille : les cellules sont donc divisées par 10, 6, 4, ou 2 selon leur normalité. A cause du manque de données, les bases ne sont pas parfaitement équilibrées, en l'occurrence les anormales (puisque l'une des bases n'avait pas les types d'anormalité) mais cela n'a pas une trop grande importance ici. A l'avenir, avec plus de données, nous pourrions équilibrer parfaitement, et peut-être même sans rotation.

Résultats

Pour comparer avec l'apprentissage pour détecter sur une image est une cellule ou non, nous avons pris les mêmes valeurs de tailles des imagerie et d'epochs. Le tout est rassemblé dans deux tableaux : le premier comporte les statistiques générales de réussite tandis que le second donne le résultat par type de cellules. Les codes couleurs est le même que pour le cas précédent.

	Taille des images en pixel	Nombre d'Epochs	Pourcentage d'anormales bien classés (P)	Pourcentage d'anormales mal classés (FN)	Pourcentage de normales bien classés (P)	Pourcentage de normales mal classés (FP)	Total Positif (P)
Test N°1	64	25	96.6	3.4	85.0	15.0	90.7
Test N°2	64	25	96.8	3.2	81.5	18.5	89.0
Test N°3	64	50	89.8	10.2	78.1	21.9	83.9
Test N°4	64	100	84.1	15.9	79.7	20.3	81.9
Test N°5	64	10	89.8	10.2	90.1	9.9	90.0
Test N°6	64	20	98.8	1.2	65.2	34.8	81.8
Test N°7	64	15	93.8	6.2	88.0	12.0	90.9
Test N°8	32	25	97.2	2.8	79.6	20.4	88.3
Test N°9	16	25	86.1	13.9	85.7	14.3	85.9
Test N°10	16	10	88.4	11.6	88.1	11.9	88.2
Test N°11	128	25	92.7	7.3	85.8	16.2	88.2
Test N°12	32	10	93.4	6.6	88.4	11.6	90.8
Test N°13	64	5	68.7	31.3	96.7	3.3	82.8
Test N°14	32	5	96.4	3.6	83.1	16.9	89.7
Test N°15	16	5	93.5	6.5	79.0	21.0	86.2
Test N°16	16	8	90.1	9.9	88.4	11.6	89.2
Test N°17	32	8	90.3	9.7	90.5	9.5	90.4
Test N°18	64	8	95.2	4.8	81.6	18.4	88.3
Test N°19	128	5	92.7	7.3	87.6	12.4	90.1
Test N°20	64	1	98.8	1.2	96.6	3.4	82.0
Test N°21	32	1	68.3	31.7	96.5	3.5	82.5
Test N°22	16	1	46.6	53.4	97.6	2.43	72.3
Test N°23	16	3	59.6	40.4	96.7	3.3	78.3
Test N°24	32	3	94.6	5.4	86.2	13.8	90.4
Test N°25	64	3	90.3	9.7	90.9	9.1	90.6
Test n°26	64	10	97.1	2.9	79.1	20.9	88.0
Test n°27	64	75	95.2	4.8	64.4	35.6	79.9
Test n°28	64	30	98.8	1.2	66.2	33.8	82.3
Test n°29	64	40	98.4	1.6	62.7	37.3	80.4
Test n°30	64	35	89.7	10.3	91.5	8.5	90.6
Test n°31	64	45	96.1	3.9	80.0	20.0	88.0
Test n°32	64	60	91.1	8.9	57.0	43.0	74.0
Test n°33	64	70	93.2	6.8	84.7	15.3	88.9
Test n°34	32	15	96.4	3.6	82.6	17.4	89.4
Test n°35	32	30	91.7	8.3	90.4	9.6	91.1
Test n°36	32	50	82.4	17.6	85.4	14.6	83.9
Test n°37	32	40	95.6	4.4	81.0	19.0	88.3

Les tests n°26 à n°37 ont été ajoutés en fonction des résultats des premiers test.
 Pour ce qui est du second tableau, la légende de couleur est indiqué ici :

Légende :	X > 95%
	95% > X > 85%
	85% > X > 80%
	80% > X > 70%
	70% > X > 60%
	X < 60%

	ANORMALES		DYSTROPHI ES		ENDOCERVIC ALES		INTERMÉDIAIR E		PARABASALE		POLYNUCLÉAI RE		SUPERFICIELL E	
	NB : 1440		NB : 1302		NB : 1336		NB : 1474		NB : 920		NB : 1136		NB : 1160	
Test N°1	97,67	2,33	69,03	30,98	83,79	16,21	91,14	8,87	82,65	17,35	97,86	2,14	95,75	4,26
Test N°2	97,93	2,08	64,15	35,86	80,20	14,29	84,66	15,35	78,94	21,06	97,40	2,61	96,14	3,87
Test N°3	91,60	8,40	66,37	33,63	79,85	20,16	84,17	15,84	73,34	26,67	95,45	4,55	70,26	29,75
Test N°4	85,06	14,94	68,99	31,02	71,09	28,91	73,68	26,33	62,29	37,86	97,67	2,33	98,82	1,19
Test N°5	93,90	6,10	79,10	20,90	91,47	8,53	95,63	4,37	82,67	17,34	98,39	1,62	99,11	0,90
Test N°6	98,84	1,16	37,01	63,00	70,15	29,86	56,59	43,42	58,27	41,73	96,15	3,85	89,39	10,61
Test N°7	95,89	4,11	73,82	26,19	88,04	11,97	94,59	5,42	82,31	17,69	98,70	1,31	97,55	2,46
Test N°8	98,00	2,00	57,32	42,69	77,10	22,90	81,22	18,79	76,10	23,90	98,30	1,70	94,82	5,18
Test N°9	91,28	8,72	67,60	32,40	81,67	18,34	88,48	11,52	80,84	19,17	98,78	1,22	98,71	1,29
Test N°10	92,15	7,61	70,11	29,89	88,40	11,60	91,46	8,54	86,02	13,98	99,46	0,55	98,85	1,16
Test N°11	94,99	5,01	65,64	34,37	86,36	13,65	88,01	11,99	78,04	21,96	98,23	1,77	94,14	5,87
Test N°12	95,93	4,07	72,80	27,20	89,14	10,87	93,26	6,74	85,23	14,77	99,46	0,55	98,01	3,46
Test N°13	76,64	23,36	88,75	11,25	95,77	4,24	99,45	0,55	95,92	4,09	99,87	0,14	99,87	0,13
Test N°14	97,77	2,24	56,26	43,75	88,59	11,41	82,28	17,73	84,34	15,67	99,09	0,92	99,04	0,97

Test N°15	96,18	3,82	48,11	51,90	81,51	18,49	76,61	23,40	77,77	22,23	98,88	1,12	96,48	3,53
Test N°16	93,90	6,10	73,24	26,77	91,98	8,03	90,92	9,09	88,21	11,79	98,86	1,14	97,12	0,26
Test N°17	94,14	5,86	80,15	19,85	87,62	12,38	95,57	4,43	85,00	15,00	99,14	0,87	98,66	1,35
Test N°18	96,90	3,10	60,79	39,21	84,85	15,16	80,75	19,26	78,29	21,71	98,03	1,98	97,31	2,69
Test N°19	95,95	4,05	64,53	35,48	94,44	5,56	88,89	11,12	87,15	12,85	99,14	0,87	98,59	1,41
Test N°20	75,75	24,25	88,18	11,82	95,96	4,04	99,28	0,73	97,81	2,19	99,87	0,14	99,94	0,07
Test N°21	77,03	22,97	88,32	11,69	97,63	2,37	99,27	0,74	98,04	1,96	99,93	0,07	100,00	0,00
Test N°22	53,57	46,43	92,71	7,29	98,48	1,52	99,41	0,60	96,79	3,21	99,93	0,07	100,00	0,00
Test N°23	67,96	32,04	89,85	10,15	95,69	4,31	99,27	0,74	96,79	3,21	100,00	0,00	99,94	0,07
Test N°24	96,72	3,28	67,18	32,83	90,78	9,22	87,48	12,53	84,25	15,75	99,34	0,66	97,69	2,32
Test N°25	94,00	6,01	77,25	22,75	94,67	5,33	94,03	5,97	88,92	11,09	99,47	0,54	99,36	0,64
Test N°26	98,07	1,93	59,82	40,18	79,83	20,17	78,41	21,60	74,21	25,79	96,99	3,02	90,07	9,94
Test N°27	97,34	2,67	45,12	54,88	45,51	54,50	71,56	28,45	53,90	46,10	88,84	11,16	73,31	26,69
Test N°28	98,98	1,02	47,34	52,67	61,37	38,64	72,13	27,88	59,00	41,00	93,89	6,11	67,91	32,10
Test N°29	98,93	1,07	31,26	68,73	70,82	29,19	59,48	40,53	60,90	39,10	97,24	2,77	65,29	34,71
Test N°30	93,53	6,47	80,77	19,23	91,71	8,30	94,70	5,30	84,31	15,69	99,54	0,47	99,74	0,26
Test N°31	96,69	3,31	66,05	33,95	71,54	28,47	85,46	14,54	74,25	25,75	98,04	1,96	95,44	4,56
Test N°32	93,94	6,06	41,97	58,04	56,08	43,92	59,27	40,73	35,59	64,42	91,69	8,32	52,76	47,24
Test N°33	95,12	4,88	55,24	44,76	88,37	11,64	84,99	15,02	83,23	16,77	99,27	0,73	97,74	2,27
Test N°34	97,64	2,36	67,12	32,88	80,91	19,09	85,75	14,26	74,88	25,13	97,19	2,81	92,40	7,60
Test N°35	94,81	5,19	76,50	23,51	90,16	9,85	94,87	5,13	87,40	12,60	99,34	0,66	97,53	2,47
Test N°36	88,50	11,50	69,55	30,46	85,08	14,93	88,70	11,30	80,10	19,90	98,56	1,44	96,96	3,04
Test N°37	97,31	2,69	59,56	40,45	76,72	23,29	84,48	15,52	70,46	29,54	98,92	1,08	96,00	4,01

Moyenne Générale	Anor	91,908	Dys	66,688	Endo	83,331	Inter	85,832	Para	79,032	Poly	98,022	Super	92,989
------------------	------	--------	-----	--------	------	--------	-------	--------	------	--------	------	--------	-------	--------

Les deux tableaux présentés ici ont été simplifiés. Pour avoir les résultats complets, il suffit d'aller sur le lien suivant : <https://drive.google.com/open?id=1N1hpHL9JcyabXMr-LFL3Eiq-845msMoxFUOWsrrk6-o>

Comme nous pouvons le constater, les résultats atteignent un score entre 85% et 90%, voir jusqu'à 91%. Il s'agit d'un très bon résultat vu la base réduite ainsi que la non prise en compte de la taille des cellules, élément crucial dans la reconnaissance des cellules anormales. En prenant en compte aussi les améliorations prévues sur l'algorithme de reconnaissance, nous pouvons être très satisfait d'avoir un tel taux de reconnaissance, qui pourra être amélioré, nous l'espérons, de quelques pourcents avec une base plus fournie.

Par contre, il est difficile de définir un modèle équilibré entre les deux, mais le choix se portera plutôt sur quelque chose qui privilégiera les cellules anormales. Le test N°35 sera sans doute préféré. Sur ces tests, nous pouvons déjà estimer que les images ne devront pas être réduites, contrairement à la reconnaissance des cellules, sous peine de perdre trop de détails et aboutir à une mauvaise reconnaissance des anormales. Aussi, nous voyons que certains types, comme les cellules dystrophiques, sont difficilement reconnus. Cela confirme ce que l'expert en cytologie avait dit : ce type là sera particulièrement compliqué à reconnaître. Nous pouvons voir un taux assez bas pour Parabasale mais il s'agit du type le moins fourni, ce qui pourrait l'expliquer. Quant à Polynucléaire et Superficielle, très différentes des autres, la reconnaissance ne pose aucun souci. Pour les autres, cela s'équilibre assez bien.

D'autres tests pourront être effectués lors d'une base plus grande, que ce soit par de multiples rotations ou plus de cellules grâce au travail d'étudiants pour améliorer encore la rapidité de fabrication des bases, voir si ces changements pourront parfaire les résultats déjà très bons sur la reconnaissance d'anormalité. En attendant, nous allons tenter de reconnaître plus précisément les types de cellules, entre Anormale et toutes les sortes de Normales.

Reconnaissance multi-types de cellules

Maintenant que nous sommes capables de différencier des cellules selon leur normalité, nous allons tenter de reconnaître précisément leur type. Si le plus gros but est de voir si les cellules sont anormales, les autres types donnent énormément d'indications sur la patiente, comme le moment où a été réalisé le prélèvement, mais aussi voir si celui-ci a bien été effectué ou surveiller certaines cellules qui seraient susceptibles de devenir anormale.

Reconnaissance générale des types

Tout d'abord, nous allons commencer avec sept classes différentes, décidées par les médecins : anormales, dystrophies, endocervicales, intermédiaires, parabasale, polynucléaire et superficielle. Il s'agit notamment des classes utilisées dans les autres travaux sur les cellules cytologiques. Comme il s'agit d'un diagnostic précis, la base de la partie précédente sera utilisée.

En ce qui concerne le deep learning, le principe sera le même. Cependant, le code actuel étant prévu pour un fonctionnement binaire, il est plus que nécessaire de le modifier afin que celui-ci accepte plus de classes. Si cela change quelques paramètres, l'utilisation n'en est que peu modifiée.

Ensuite, afin de tester l'efficacité de l'apprentissage, nous calculerons plusieurs choses. D'abord, nous compterons, comme les autres tests, le nombre de cellules bien reconnues et mal reconnues. Ensuite, nous nous intéresserons au type trouvé à chaque cellule. En effet, il est intéressant de voir quel était l'erreur de l'algorithme afin de pouvoir constater quelles classes se confondent le plus. Le tout sera séparé selon les classes d'origine afin de pouvoir voir celles qui sont les mieux reconnues par rapport aux autres. Comme précédemment, le tout est rangé dans autant de fichiers CSV qu'il y a de types.

Comme il n'est pas possible de faire tenir autant d'informations dans un seul tableau, nous utiliserons deux tableaux : un qui donnera les statistiques générales sur le cas traité puis par type et un autre qui donnera, pour chaque classe, le type que l'algorithme lui a donné et le nombre de cellules trouvées dans chaque classe. Ce dernier donnant un tableau par cas, nous n'allons mettre que certains exemples.

Voici donc le premier tableau obtenu :

				ANORMALES	DYSTROPHIES	ENDOCERVICALES	INTERMÉDIAIRE	PARABASALE	POLYNUCLÉAIRE	SUPERFICIELLE							
	Taille	Ech	Total Positif (P)	NB : 1440	NB : 1302	NB : 1336	NB : 1474	NB : 920	NB : 1136	NB : 1160							
Test N°1	64	25	56.8	82,82	17,18	2,72	97,28	35,33	64,68	53,19	46,82	48,69	51,31	96,23	3,77	57,90	42,11
Test N°2	64	25	51.9	75,30	24,70	44,21	55,79	26,09	73,91	58,08	41,92	15,40	84,60	97,05	2,96	7,46	92,55
Test N°3	64	50	45.1	89,78	10,22	17,52	82,47	1,21	98,79	68,13	31,87	12,10	87,90	92,92	7,08	0,78	99,22
Test N°7	64	15	58.1	87,25	12,75	1,84	98,16	35,78	64,22	67,21	32,79	31,90	68,10	97,08	2,92	57,08	42,92

Test N°8	32	25	43.8	98,94	1,06	0,05	99,96	67,35	32,66	3,74	96,26	11,06	88,94	61,74	38,26	38,17	61,83
Test N°9	16	25	34.0	96,14	3,86	0,22	99,78	24,82	75,19	0,00	100	6,25	93,75	99,39	0,62	1,10	98,90
Test N°11	12	25	41.7	98,67	1,34	0,09	99,91	12,84	87,16	4,63	95,38	1,50	98,50	88,95	11,06	68,83	31,18
Test N°13	64	5	50.8	56,33	43,68	3,73	96,27	18,44	81,56	74,86	25,14	22,15	77,85	96,69	3,31	71,22	28,78
Test N°18	64	8	58.6	96,20	3,80	14,01	85,99	46,91	53,09	46,37	53,63	28,25	71,75	95,62	4,38	69,54	30,46
Test N°25	64	3	56.4	72,86	27,14	12,29	87,71	23,91	76,09	65,67	34,33	30,02	69,98	95,66	4,34	72,48	27,53
Test N°26	64	10	62.2	89,30	10,70	0,79	99,21	39,70	27,80	49,73	50,27	38,77	61,23	92,85	7,16	65,49	34,51
Test N°27	64	75	37.1	99,95	0,05	5,13	94,87	13,83	86,17	8,27	91,74	3,06	96,94	75,91	24,09	34,25	65,76
Test N°28	64	30	52.1	86,55	13,45	20,93	79,08	72,14	27,87	33,11	66,89	18,88	81,13	93,10	6,90	7,21	92,79
Test N°30	64	35	51.9	89,74	10,26	4,17	95,83	12,05	87,96	73,69	26,31	34,25	65,75	88,90	11,11	32,85	67,15
Test N°38	64	12	64.0	84,97	15,03	7,50	92,50	41,21	58,79	70,62	29,38	53,52	46,48	87,29	12,72	76,51	23,49
Test N°39	64	14	61.5	71,06	28,94	21,95	78,06	46,32	53,69	70,60	29,40	35,59	64,42	95,39	4,62	54,52	45,48
Test N°40	64	9	58.7	78,50	21,50	6,06	93,95	48,55	51,46	50,86	49,14	49,79	50,21	96,63	3,37	68,95	31,06
Test N°41	64	11	62.1	83,34	16,66	5,58	94,43	49,87	50,13	75,63	24,37	28,67	71,34	96,66	3,34	58,87	41,13
Test N°42	64	13	64.2	84,74	15,26	2,41	97,59	52,17	47,84	72,65	27,35	55,10	44,90	93,59	6,41	71,74	28,26

Moyen																	
ne																	
Général	Anor	85,392	Dys	9,009	Endo	35,183	Inter	49,843	Para	27,629	Poly	91,664	Super	48,154			

Ce tableau est une version simplifiée. La légende des couleurs est la même que pour la détection entre normale et anormale. Vous pouvez retrouver le tableau complet sur le lien suivant pour des informations plus précises : https://drive.google.com/open?id=1dEpWZhsuaLkb3u9NQjZmvQOPdJh0ipbiB3CFuop_cto

Pour compléter, voici quelques exemples des tableaux qui récapitule, pour chaque test, la classification générale et comptée de chaque cellule dans chaque type. Le premier donne un exemple d'une mauvaise classification et le second sur la meilleure classification :

Test N°3	Type originelle	Reconnues Majoritairement comme	Nombre reconnues par type	Anormale	Dystrophiques	Endocervicale	Intermédiaire	Parabasale	Polynucléaire	Superficielle
	Anormale	Anormale		1348	43	0	39	5	5	0
	CytoPro (Anormale)	Anormale		940	43	0	25	4	2	0
	ASCUS (CellIA)	Anormale		25	0	0	6	1	0	0
	HSIL (CellIA)	Anormale		305	0	0	0	0	1	0
	LSIL (CellIA)	Anormale		78	0	0	8	0	2	0
	Dystrophie	Anormale		534	345	0	287	26	110	0
	CellIA	Intermédiaire		35	9	0	63	1	54	0
	CytoPro	Anormale		499	336	0	224	25	56	0
	Endocervicale	Anormale		813	151	31	43	1	297	0
	CellIA	Anormale		40	1	0	3	0	12	0
	CytoPro	Anormale		773	150	31	40	1	285	0
	Intermédiaire	Intermédiaire		210	90	0	1017	8	149	0
	CellIA	Intermédiaire		54	31	0	256	2	43	0
	CytoPro	Intermédiaire		156	59	0	761	6	106	0
	Parabasale	Polynucléaire		181	217	3	127	154	238	0
	CellIA	Anormale		72	17	3	12	7	9	0
	CytoPro	Polynucléaire		109	200	0	115	147	229	0
	Polynucléaire	Polynucléaire		72	1	2	2	8	1051	0
	CellIA	Polynucléaire		22	0	0	0	1	381	0
	CytoPro	Polynucléaire		50	1	2	2	7	670	0
	Superficielle	Polynucléaire		438	18	1	63	4	627	9
	CellIA	Anormale		177	17	1	51	0	131	3
	CytoPro	Polynucléaire		261	1	0	12	4	496	6

Test N°42	Type originelle	Reconnues Majoritairement comme	Nombre reconnues par type	Anormale	Dystrophiques	Endocervicale	Intermédiaire	Parabasale	Polynucléaire	Superficielle
	Anormale	Anormale		1272	19	0	85	53	1	10
	CytoPro (Anormale)	Anormale		874	17	0	67	46	0	10
	ASCUS (CellIA)	Anormale		21	0	0	9	2	0	0

	HSIL (CellIA)	Anormale	298	1	0	3	3	1	0
	LSIL (CellIA)	Anormale	79	1	0	6	2	0	0
	Dystrophie	Parabasale	405	55	23	374	409	21	15
	CellIA	Intermédiaire	24	0	0	81	40	17	0
	CytoPro	Anormale	381	55	23	293	369	4	15
	Endocervicale	Endocervicale	129	9	636	110	234	218	0
	CellIA	Endocervicale	5	0	32	5	11	3	0
	CytoPro	Endocervicale	124	9	604	105	223	215	0
	Intermédiaire	Intermédiaire	79	3	30	1128	103	107	24
	CellIA	Intermédiaire	22	0	27	249	49	17	22
	CytoPro	Intermédiaire	57	3	3	879	54	90	2
	Parabasale	Parabasale	63	13	77	153	587	27	0
	CellIA	Parabasale	16	0	45	4	52	3	0
	CytoPro	Parabasale	47	13	32	149	535	24	0
	Polynucléaire	Polynucléaire	9	0	23	7	35	1056	6
	CellIA	Polynucléaire	3	0	0	3	7	387	4
	CytoPro	Polynucléaire	6	0	23	4	28	669	2
	Superficielle	Superficielle	19	0	4	124	18	99	896
	CellIA	Superficielle	0	0	2	99	18	49	212
	CytoPro	Superficielle	19	0	2	25	0	50	684

Il s'agit de quelques exemples des tableaux qui récapitule les types. Pour voir tous les tableaux pour chaque test, il faut cliquer sur ce lien : <https://drive.google.com/open?id=1XZ0qmMRq3DO0hTwHtzEN1pLDeY7cht7ZqPF3-z7f8tM>

Comme nous pouvons le constater, les résultats ne sont pas excellents. Nous avons, au mieux 64,2% de reconnaissance générale, ce qui n'est clairement pas suffisant. De plus, certains types sont mieux reconnus que d'autres en général : Anormales et Polynucléaire s'en sortent bien, jusqu'à parfois absorber les autres classe tandis que Dystrophies a un taux catastrophique de reconnaissance, n'excédant pas 10% en moyenne.

De manière générale, les confusions de types s'expliquent : les dystrophies sont reconnues comme anormales car assez proche. Pareil pour les classes Endocervicale et Intermédiaire. Comme les différences ne sont pas suffisantes, l'algorithme les confond. Cependant, quelques incompréhensions subsistent, notamment dans la reconnaissance moyenne des Superficielles, alors qu'il s'agit d'une classe bien distincte, ou alors la confusion des autres types avec Polynucléaire, pourtant bien unique en terme de forme.

Il est cependant difficile de conclure sur cet apprentissage. En effet, nous n'avons qu'un millier de cellules dans chaque classe, parfois moins comme pour Parabasale. En terme de deep learning, c'est assez peu, alors que la base a été agrandie par des rotations. Par rapport à une détection entre cellules et non cellules, cela en fait 50 fois moins et 7 fois

moins par rapport à l'apprentissage entre normales et anormales. De fait, il est tout à fait possible que le manque de données puisse être la cause de ces résultats médiocres, puisque l'abondance de données est un gardien du bon fonctionnement de l'algorithme. Pour améliorer ce point, nous allons à la fois essayer de récupérer plus de cellules grâce au travail de François Lozes et Michel Lécluse sur CellIA, mais aussi essayer d'effectuer plus de rotations sur la base, voir à quel moment peut survenir un sur-apprentissage sur les mêmes données.

Mais, pour le moment, nous allons tenter une autre option : réduire le nombre de classes pour séparer efficacement les cellules selon leur véritable type. D'abord, nous allons enlever la classe des Anormales, vu que nous avons quelque chose d'efficace pour les détecter, et ne séparer que les Normales. Par ce biais, nous pouvons espérer un meilleur résultat. Puis, nous testerons à la fois d'injecter des caractéristiques précises à l'apprentissage pour aider l'algorithme (comme sur la surface d'une cellule ou sa densité) mais aussi de multiplier les apprentissages, de manière binaire, afin de séparer petit à petit chaque cellule à un paquet de classes et améliorer ainsi la reconnaissance (par exemple séparer Normales et Anormales, puis Dystrophies et le reste, etc).

Reconnaissance des types des cellules normales

Comme nous avons pu le constater, la détection générale des types n'est pas suffisante à l'heure actuelle. Parmi les classes qui peuvent fausser le résultat, nous avons les anormales. Au vue des résultats dans la partie précédente sur la reconnaissance entre normales et anormales, nous pouvons imaginer une détection en deux étapes : d'abord séparer les normales des anormales pour ensuite détecter les types. De fait, nous n'aurons plus 7 classes mais 6.

Qu'il s'agisse de l'apprentissage, la base de données ou bien des résultats, nous allons utiliser sensiblement les mêmes choses que précédemment. La seule différence réside dans le retrait d'une classe. Ce travail étant en cours, nous allons présenter les tableaux à l'heure actuelle, sans pour autant poser toutes les conclusions.

				DYSTROPHI ES		ENDOCERVIC ALES		INTERMÉDIAIR E		PARABASALE		POLYNUCLÉAI RE		SUPERFICIELL E	
	Ta ille	Epo chs	Total Positif (P)	NB : 1302		NB : 1336		NB : 1474		NB : 920		NB : 1136		NB : 1160	
Test N°1	64	25	65.9	44,29	55,72	65,58	34,43	50,82	49,18	33,00	67,00	90,29	9,72	72,31	27,69
Test N°2	64	25	56.7	15,89	84,11	49,40	50,60	65,17	34,84	24,60	75,40	96,53	3,48	49,37	50,63
Test N°3	64	50	57.1	49,34	50,66	70,20	29,81	65,73	34,27	5,63	94,38	89,01	10,99	17,63	82,37
Test N°4	64	100	31.9	25,37	74,64	1,56	98,44	91,67	8,34	0,00	100,00	35,81	64,19	10,95	89,05
Test N°7	64	15	63.4	17,12	82,88	45,90	54,11	66,10	33,90	52,92	47,09	95,47	4,54	72,94	27,07

Test N°8	32	25	43.8	41,07	58,94	19,83	80,17	12,63	87,37	23,09	76,92	98,71	1,29	38,83	61,18
Test N°9	16	25	35.8	19,68	80,33	97,03	2,97	0,00	100,00	8,13	91,88	77,08	22,93	0,99	99,02
Test N°11	12	25	63.5	29,62	70,39	81,43	18,57	43,32	56,68	31,48	68,52	92,66	7,34	70,48	29,52
Test N°13	64	5	63.8	14,48	85,52	40,74	59,26	78,18	21,83	52,71	47,29	93,59	6,41	76,44	23,56
Test N°18	64	8	59.3	12,69	87,32	55,17	44,83	56,61	43,40	48,00	52,00	97,48	2,52	71,01	29,00
Test N°25	64	3	56.2	18,09	81,91	20,76	79,24	69,06	30,94	33,85	66,15	95,93	4,07	73,50	26,51
Test N°26	64	10	66.8	31,48	68,53	66,56	33,44	79,37	20,63	21,85	78,15	94,62	5,38	62,63	37,37
Test N°27	64	75	37.1	26,93	73,07	66,72	33,29	13,28	86,72	0,13	99,88	91,80	8,20	7,85	92,15
Test N°28	64	30	68.1	20,41	79,59	68,87	31,13	74,03	25,98	40,50	59,50	93,05	6,96	71,62	28,39
Test N°29	64	40	47.5	47,23	52,78	41,27	58,74	23,53	76,48	10,94	89,06	98,16	1,84	42,78	57,22
Test N°30	64	35	54.8	55,11	44,90	77,39	22,62	14,15	85,85	1,00	99,00	92,73	7,27	61,19	38,82
Test N°38	64	12	66.3	17,92	82,09	47,25	52,75	82,74	17,26	56,59	43,42	94,48	5,52	73,56	26,44
Test N°39	64	14	65.2	18,18	81,83	38,72	61,29	74,96	25,05	58,96	41,04	93,57	6,43	70,27	29,74
Test N°40	64	9	57.5	14,92	85,08	20,67	79,34	71,07	28,94	59,92	40,09	96,40	3,60	67,49	32,51
Test N°41	64	11	53.7	16,64	83,37	17,15	82,85	67,41	32,59	27,73	72,27	98,22	1,78	67,24	32,77
Test N°42	64	13	63.3	33,14	66,87	67,47	32,54	42,60	57,40	46,40	53,60	94,62	5,38	59,82	40,18

Moyenne Générale	Dys	27,121	Endo	50,458	Inter	54,400	Para	30,352	Poly	90,961	Super	54,231
------------------	-----	--------	------	--------	-------	--------	------	--------	------	--------	-------	--------

Ce tableau est une version simplifiée et non terminée. La légende des couleurs est la même que pour la détection entre normale et anormale. Vous pouvez retrouver le tableau complet sur le lien suivant pour des informations plus précises :

<https://drive.google.com/open?id=1UdD-HYIVK2AGwpClgobsSTx4pm-cNEmEzEhtilK4wGo>

De la même manière que précédemment, nous allons mettre deux exemples de tableaux qui récapitule les reconnaissances dans chaque classe :

Test N°27	Type originelle	Reconnues Majoritairement comme	Nombre reconnues par type	Dystrophies	Endocervicale	Intermédiaire	Parabasale	Polynucléaire	Superficielle
	Dystrophie	Polynucléaire		439	306	74	0	477	6
	CellIA	Polynucléaire		29	16	6	0	111	0
	CytoPro	Dystrophie		410	290	68	0	366	6
	Endocervicale	Endocervicale		64	921	4	0	347	0
	CellIA	Endocervicale		5	36	0	0	15	0
	CytoPro	Endocervicale		59	885	4	0	332	0
	Intermédiaire	Polynucléaire		214	102	229	0	929	0
	CellIA	Polynucléaire		57	35	33	0	261	0
	CytoPro	Polynucléaire		157	67	196	0	668	0
	Parabasale	Polynucléaire		208	212	5	2	493	0
	CellIA	Endocervicale		0	102	0	0	18	0
	CytoPro	Polynucléaire		208	110	5	2	475	0
	Polynucléaire	Polynucléaire		5	98	0	0	1033	0
	CellIA	Polynucléaire		1	20	0	0	383	0
	CytoPro	Polynucléaire		4	78	0	0	650	0
	Superficielle	Polynucléaire		95	34	16	0	921	94
	CellIA	Polynucléaire		42	23	16	0	272	27
									67
	CytoPro	Polynucléaire		53	11	0	0	649	

Test N°28	Type originelle	Reconnues Majoritairement comme	Nombre reconnues par type	Dystrophies	Endocervicale	Intermédiaire	Parabasale	Polynucléaire	Superficielle
	Dystrophie	Dystrophie		405	90	390	276	102	39
	CellIA	Intermédiaire		10	0	81	4	64	3
	CytoPro	Dystrophie		395	90	309	272	38	36
	Endocervicale	Endocervicale		51	1020	40	58	167	0
	CellIA	Endocervicale		4	34	4	6	8	0
	CytoPro	Endocervicale		47	986	36	52	159	0
	Intermédiaire	Intermédiaire		70	20	1127	57	170	30
	CellIA	Intermédiaire		19	9	266	24	48	20
	CytoPro	Intermédiaire		51	11	861	33	122	10
	Parabasale	Parabasale		71	160	120	495	74	0
	CellIA	Endocervicale		0	93	0	27	0	0
	CytoPro	Parabasale		71	67	120	468	74	0
	Polynucléaire	Polynucléaire		1	57	11	11	1048	8
	CellIA	Polynucléaire		1	5	2	3	387	6
	CytoPro	Polynucléaire		0	52	9	8	661	2

	Superficielle	Superficielle		7	20	99	18	122	894
	CellIA	Superficielle		5	9	76	17	61	212
	CytoPro	Superficielle		2	11	23	1	61	682

Il s'agit de quelques exemples des tableaux qui récapitule les types. Pour voir tous les tableaux pour chaque test, il faut cliquer sur ce lien : https://drive.google.com/open?id=1WOH90NinmIQBO6A-_3Y1MQAjjCVtG5KIRPi0S5MPg48

Aux premiers abords, nous pouvons voir un meilleur taux, surtout pour les cellules dystrophiques. Seules les cellules parabasales semblent être moins reconnues, mais il s'agit de la classe qui possède le moins de données, ce qui n'aide pas à la reconnaissance. Globalement, il est possible que le taux général soit meilleur qu'avant. Aussi, avec les divers améliorations citées auparavant, comme l'augmentation de données, nous pouvons aussi espérer un meilleur taux.

Cependant, si nous voulons un résultat et des statistiques corrects, il faut prendre en compte le traitement fait au préalable, soit la reconnaissance entre les cellules anormales et normales. Pour faire les calculs, nous allons considérer que nous avons 90% de reconnaissances des Anormales et de même pour la reconnaissance des Normales. Dans notre cas, sur le nombre de Normales établies, nous avons 10% manquant et c'est sur cette partie que nous classons par type. Ce qui donne le résultat suivant :

Proba que la cellule soit reconnue comme normale	0,904
Proba que la cellule soit reconnue comme anormale	0,917

	10 rotations max	20 rotations max	40 rotations max
Avec Anormales	64,2	68,3	67,5
Sans Anormales	68,1	67,1	69,5
Vrai Proba Sans Anormales	58,9	57,5	59,9

Comme nous pouvons le voir, nous avons gardé les meilleurs résultats possibles selon le nombre de rotations, point que nous allons aborder plus bas dans ce rapport. Si les taux sont légèrement meilleurs lorsque nous séparons les cellules Anormales des Normales, nous pouvons constater que cela n'est pas suffisant lors de la combinaison des deux. En effet, les 90% entraînent une trop grosse perte, et donc réduit énormément le pourcentage associé. Si le calcul n'est pas exact vu que nous n'avons pas tenté de cette manière, nous ne pourrions avoir de meilleurs résultats puisque 10% des Anormales feront des erreurs dans les Normales et vice-versa. De fait, il n'est malheureusement pas intéressant de séparer les Normales, même si nous avons individuellement de meilleurs reconnaissances.

Cependant, si ce taux n'est malheureusement pas encore suffisant pour être exploité, il peut sans doute se renforcer. Pour ce faire, nous allons essayer d'augmenter le nombre de rotations maximum, afin de voir si cet ajout de données aide à la reconnaissance ou si la redondance inhibe l'apprentissage.

Augmentation de la base de données par rotation

Comme nous l'avons signalé, nous avons effectué des rotations sur les cellules afin d'agrandir la base de données, puisque l'ordinateur considère qu'il s'agit d'objets différents. Si beaucoup utilisent cette technique sur des objets qui n'ont pas de sens, il n'y a pas vraiment de recherches sur jusqu'où il est possible d'aller sans compromettre la base de données avec des redondances. C'est ce que nous allons faire ici.

Si la base de données acceptaient jusqu'à 10 rotations maximum, ici nous allons essayer d'en réaliser 20 puis 40, voir ce que cela donne. Bien entendu, nous garderons toujours l'équilibre de la base. Si nous reprenons le précédent tableau, récapitulant le nombre de cellules selon chaque type, dans le cas binaire Anormal/Normal, nous obtiendrons donc ceci, avec le nombre de rotations indiquées.

	Base CellIA	Base CytoPro	Total	Coef 10 Rotations		Coef 20 Rotations		Coef 40 rotations	
ANORMAL	428	1014	1442	2	2884	4	5768	8	11536
ASCUS	33	-	-		66				
HSIL	307	-	-		614				
LSIL	88	-	-		176				
NORMAL	1279	3783	5062		14678		29356		
Dystrophies	55	381	436	6	2616	12	5232	24	10464
Endocervicale	29	640	669	4	2676	8	5352	16	10704
Intermédiaire	386	1089	1475	2	2950	4	5900	8	11800
Parabasale	24	160	184	10	1840	20	3680	40	7360
Polynucléaire	405	733	1138	2	2276	4	4552	8	9104
Superficielle	380	780	1160	2	2320	4	4640	8	9280
TOTAL	1707	4797	6504						

Si, à partir de 40 rotations, l'oeil humain ne voit pas tant de différences entre deux rotations proches, nous espérons que l'ordinateur en trouvera. L'apprentissage, ainsi que les tests, resteront les mêmes. Afin d'éviter d'encombrer le rapport avec 6 tableaux de plus, nous allons résumer les trois meilleurs résultats ci-dessous.

Avec le type Anormal

				ANORMAL ES		DYSTROPH IES		ENDOCERVI CALES		INTERMEDIA IRE		PARABASA LE		POLYNUCLE AIRE		SUPERFICIEL LE	
Max 10 Rotat ions	Tai lle	Ep och s	Total Positif (P)	NB : 1440		NB : 1302		NB : 1336		NB : 1474		NB : 920		NB : 1136		NB : 1160	
Test N°26	64	10	62.2	89,3	10,7	0,79	99,21	39,7	27,8	49,73	50,27	38,77	61,23	92,85	7,16	65,49	34,51
Test N°38	64	12	64.0	84,9	15,03	7,5	92,5	41,21	58,79	70,62	29,38	53,52	46,48	87,29	12,72	76,51	23,49
Test N°42	64	13	64.2	84,7	15,26	2,41	97,59	52,17	47,84	72,65	27,35	55,1	44,9	93,59	6,41	71,74	28,26
				ANORMAL ES		DYSTROPH IES		ENDOCERVI CALES		INTERMEDIA IRE		PARABASA LE		POLYNUCLE AIRE		SUPERFICIEL LE	
Max 20 Rotat ions	Tai lle	Ep och s	Total Positif (P)	NB : 3012		NB : 2736		NB : 2672		NB : 2944		NB : 1840		NB : 2284		NB : 2324	
Test N°13	64	5	65.2	84,1	15,85	26,3	73,64	38,21	61,8	78,44	21,57	31,12	68,89	92,56	7,44	73,99	26,02
Test N°26	64	10	68.3	89,5	10,46	24,9	75,09	59,45	40,55	72,93	27,08	34,22	65,78	94,71	5,29	72,77	27,24
Test N°39	64	14	66.2	76,3	23,62	51,5	48,51	50,1	49,91	61,8	38,21	26,19	73,81	92,74	7,26	79,05	20,96
				ANORMAL ES		DYSTROPH IES		ENDOCERVI CALES		INTERMEDIA IRE		PARABASA LE		POLYNUCLE AIRE		SUPERFICIEL LE	
Max 40 Rotat ions	Tai lle	Ep och s	Total Positif (P)	NB : 6024		NB : 5448		NB : 5344		NB : 5896		NB : 3680		NB : 4568		NB : 4648	
Test N°2	64	25	67.5	71,9	28,1	30,5	69,45	70,97	29,04	60,16	39,85	45,65	54,35	92,29	7,71	65,58	34,43
Test N°18	64	8	65.2	87,1	12,91	21,0	78,92	52,91	47,09	69,17	30,83	33,22	66,78	94,53	5,48	76,09	23,91
Test N°26	64	10	64.3	62,5	37,44	52,7	47,26	74,59	25,41	37,87	62,14	29,91	70,1	91,63	8,38	71,29	28,71

Sans le type Anormal

				DYSTROPHIES	ENDOCERVICALES	INTERMEDIAIRE	PARABASALE	POLYNUCLEAIRE	SUPERFICIELLE
Max 10 Rotations	Taille	Epochs	Total Positif (P)	NB : 1302	NB : 1336	NB : 1474	NB : 920	NB : 1136	NB : 1160
Test N°26	64	10	66.8	31,48 68,53	66,56 33,44	79,37 20,63	21,85 78,15	94,62 5,38	62,63 7,37
Test N°28	64	30	68.1	20,41 79,59	68,87 31,13	74,03 25,98	40,5 59,5	93,05 6,96	71,62 28,39
Test N°38	64	12	66.3	17,92 82,09	47,25 52,75	82,74 17,26	56,59 43,42	94,48 5,52	73,56 26,44
				DYSTROPHIES	ENDOCERVICALES	INTERMEDIAIRE	PARABASALE	POLYNUCLEAIRE	SUPERFICIELLE
Max 20 Rotations	Taille	Epochs	Total Positif (P)	NB : 2736	NB : 2672	NB : 2944	NB : 1840	NB : 2284	NB : 2324
Test N°39	64	14	64.8	33,75 66,25	40,45 59,56	81 19,01	23,6 76,4	96,25 3,76	69,42 30,59
Test N°41	64	11	67.1	45,22 54,78	64,48 35,53	63,54 36,46	19,53 80,47	96,19 3,82	72,75 27,25
Test N°42	64	13	65.0	34,06 65,95	36,94 63,07	66,21 33,79	52,02 47,98	96,29 3,71	72,48 27,52
				DYSTROPHIES	ENDOCERVICALES	INTERMEDIAIRE	PARABASALE	POLYNUCLEAIRE	SUPERFICIELLE
Max 40 Rotations	Taille	Epochs	Total Positif (P)	NB : 5448	NB : 5344	NB : 5896	NB : 3680	NB : 4568	NB : 4648
Test N°1	64	25	68.5	60,08 39,93	63,97 36,03	52,71 47,3	26,53 73,48	92,27 7,73	70,87 29,14
Test N°7	64	15	66.0	59,03 40,98	60,85 39,16	47,83 52,18	43,42 56,58	94,51 5,49	62,05 37,95
Test N°26	64	10	69.5	41,03 58,98	56,88 43,12	66,71 33,3	40,34 59,66	95,57 4,44	72,96 27,04

Chaque tableau complet peut cependant être retrouvé. Voici donc la liste des liens pour accéder à chaque tableau.

Pour la reconnaissance des types avec Anormale :

- Maximum 10 Rotations :
 1. Tableau récapitulatif des résultats : https://drive.google.com/open?id=1dEpWZhsuaLkb3u9NQjZmvQOPdJh0ipbiB3CFuop_cto
 2. Tableau des reconnaissances de chaque type : <https://drive.google.com/open?id=1XZ0qmMRq3DO0hTwHtzEN1pLDeY7cht7ZqPF3-z7f8tM>
- Maximum 20 Rotations :
 1. Tableau récapitulatif des résultats : <https://drive.google.com/open?id=1w4K2lwPRcWaxKF9yaaxNy30QAZnZZOOfsxixrHfVUuc>
 2. Tableau des reconnaissances de chaque type : https://drive.google.com/open?id=1Gx1NnakQOCl5nsi9tHo5-nWG8SjrQrsesq_eLrWUhBc
- Maximum 40 Rotations :
 1. Tableau récapitulatif des résultats : <https://drive.google.com/open?id=17sd58udmiGcJKxRx8ee4sGWawEO5MjIUul9x4EmyKxM>
 2. Tableau des reconnaissances de chaque type : https://drive.google.com/open?id=1WOaC_cnMc06tbKY0Gd7BVmJg2R-iw6dfACqZQKLUvf8

Pour la reconnaissance des types sans Anormale :

- Maximum 10 Rotations :
 3. Tableau récapitulatif des résultats : <https://drive.google.com/open?id=1UdD-HYIVK2AGwpClgobsSTx4pm-cNEmEzEhtiK4wGo>
 4. Tableau des reconnaissances de chaque type : https://drive.google.com/open?id=1WOH90NlnmlQBO6A-_3Y1MQAjjCVtG5KIRPi0S5MPg48
- Maximum 20 Rotations :
 3. Tableau récapitulatif des résultats : <https://drive.google.com/open?id=1AvqAG3bMaa3xNIRJ2a69PUFyJeml8cTvtTs-VZHjTBA>
 4. Tableau des reconnaissances de chaque type : https://drive.google.com/open?id=19k7ZWsr3S3ECj1_94uA7yzMPRpTJxhZtJ0HdTtO4KLS
- Maximum 40 Rotations :
 3. Tableau récapitulatif des résultats : <https://drive.google.com/open?id=1Uhr1VafNe2jYGdH9vWrGBidLdjqlOf1mgphQkwqa1OA>
 4. Tableau des reconnaissances de chaque type : <https://drive.google.com/open?id=1wLUA3YdV2HdUTy6g71x74DWwAzS9IRBZZuSYo8Cfs3Q>

A la vue des résultats, nous pouvons constater que l'apport est minime. Si nous gagnons quelques points dans les pourcentages pour le cas prenant Anormale en compte, ce n'est pas valable pour la reconnaissance des types de cellules Normales. Cependant, aller jusqu'à 40 rotations n'est pas utile. Si cela ne ruine pas l'algorithme aux premiers abords, cela n'apporte rien de plus. Nous pouvons donc en conclure que si nous pouvons augmenter les rotations, cela n'est pas forcément utile (mise à part ajouter des données, ce qui n'est pas négatif en soi) et cela n'apporte, au bout d'un moment, aucun bénéfice. Aller jusqu'à 20 rotations paraît une bonne option, si le besoin se fait sentir. Malheureusement, cela n'apporte pas une évolution majeure dans la reconnaissance, ce qu'on aurait pourtant bien apprécié.

Comme cette technique ne fonctionne malheureusement pas suffisamment, il faudrait donc essayer d'améliorer la reconnaissance des types avec plus de données, donc plus de cellules (comme répété maintes fois) mais aussi en injectant des caractéristiques sur les cellules comme la taille, la texture ou l'environnement. Aussi, beaucoup plus d'apprentissage binaires pour séparer les classes une par une peut être une voie, mais il faudrait en discuter avec les experts en cytopathologie afin de définir les meilleurs comparaisons pour discriminer au mieux les cellules entre elles selon leur type.

En attendant, ces algorithmes peuvent s'intégrer à un traitement plus général sur une tuile, ainsi qu'à un affichage, afin de montrer facilement une tuile traitée à un utilisateur, autant sur la détection de cellules que la détection de leur anormalité et leur type.

Détection des types de cellules sur une tuile

Nous avons pu voir, à la section Détection de cellules sur une tuile, dans la partie Détection grâce aux résultats, que nous avons créé une petite interface afin de pouvoir lancer la détection et la classification des cellules sur une tuile. Nous allons présenter une version améliorée, comprenant la reconnaissance des types de cellules et leur affichage sur une tuile.

Tout d'abord, tout ce qui a été dit dans cette section est utilisée dans cette détection. En effet, il s'agit d'une étape supplémentaire : sur les cellules détectées, au lieu de les afficher, nous allons effectuer la classification de types et afficher ce nouveau résultat. De fait, les premiers étapes sont exactement les mêmes.

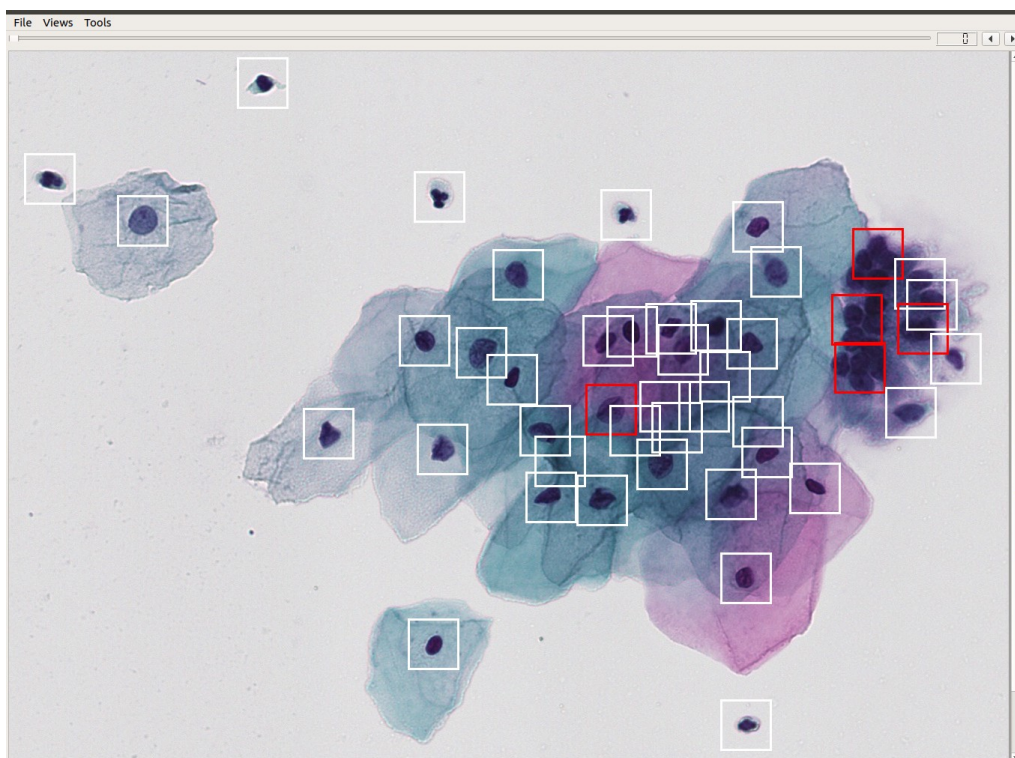
Ensuite, une fois que nous avons les imagerie et leur classification, nous avons tout ce qui nous faut pour classer par type. Le script va être assez simple : nous allons utiliser le CSV créé par l'étape précédente qui contient, rappelons le, le nom et le chemin de chaque imagerie, avec sa lame, sa tuile, sa position et sa classification (cellule ou non). De ce CSV, nous allons le parcourir selon la tuile à traiter afin de classer par type seulement les cellules.

De là, nous pourrions choisir deux algorithmes :

- Soit la version binaire (Anormale et Normale)
- Soit la version multiple (Tous les types de cellules)

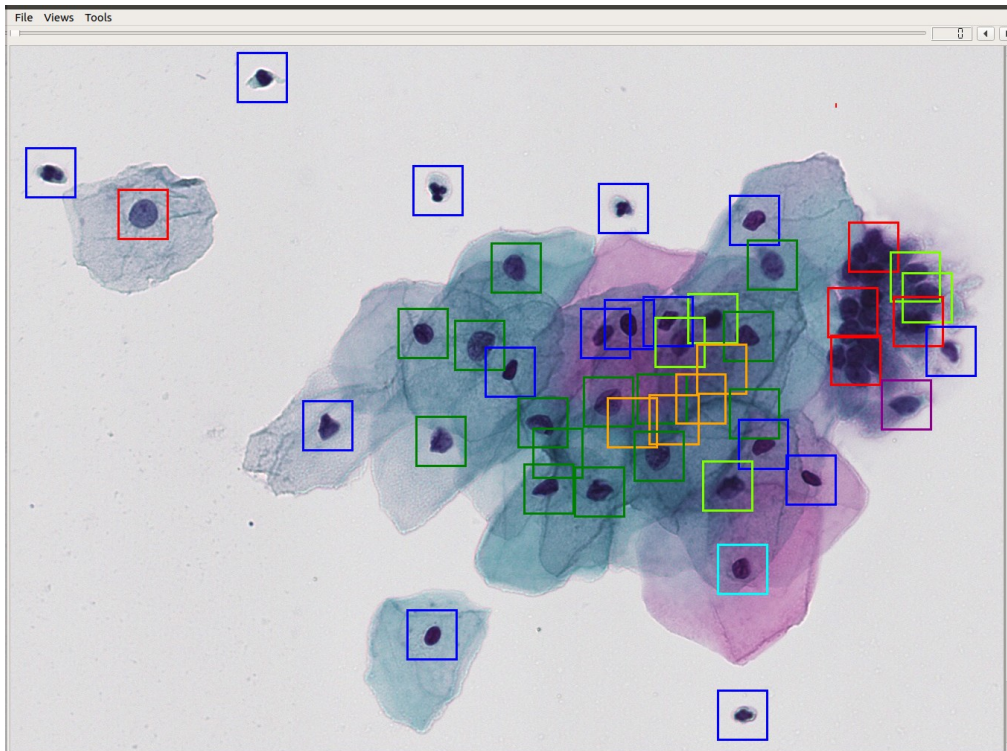
Le résultat de la classification choisie sera mis dans un nouveau CSV, qui comportera le chemin pour accéder à l'imagette, la lame à laquelle elle appartient, la tuile à laquelle elle appartient, son nom, sa position dans la tuile et son type. De fait, ce CSV ne contiendra que les résultats des cellules. Pour chacun des deux apprentissages, nous allons prendre celui qui a donné le meilleur résultat, soit le test N°35 pour l'algorithme binaire et le test N°42 pour l'algorithme multiple.

Enfin, l'affichage ne sera pas si différent de la version précédente. Pour le cas d'une classification binaire entre Anormale et Normale, nous afficherons ainsi :



Les carrés rouges indiquent les cellules dites anormales et les blancs les cellules dites normales.

Dans le cas d'une classification par type multiples, nous obtiendrons ceci :



Ici, le code couleur est plus complexe. Celui-ci est commun à CellIA, afin que les médecins ne changent pas leurs habitudes. Nous avons donc :

- Anormales en **rouge vermeil**
- Dystrophies en **orange miel**
- Endocervicale en **vert fluo**
- Intermédiaire en **vert forêt**
- Parabasale en **violet**
- Polynucléaire en **bleu foncé**
- Superficielle en **turquoise**

Comme la version précédente, plusieurs options sont disponibles. Nous pouvons choisir de traiter une ou plusieurs tuiles, avec affichage en lançant toutes les étapes. Il est aussi possible de ne faire que l'affichage si les classifications ont été faites avant. Et de même, il est possible de faire le traitement sur toute une lame. Cependant, cette option n'a pas été testée au vue des soucis dans la version précédente avec le disque dur externe. Nous pouvons compter, en tout cas, environ 2 heures de plus à la louche. Aussi, il n'est pas possible, à l'heure actuelle, de ne faire que la classification par type. Il n'est pas impossible de la lancer via le script Python, mais cela n'est pas prévu par le script général. Ce choix a été fait puisque l'étape de détection de cellules n'est qu'une étape en soi, et non pas une finalité intéressante. Cependant, un utilisateur avisé pourra sans problème trouver la fonction et les paramètres adéquats si le besoin se fait sentir.

Une option a été ajoutée : il est possible de pouvoir gagner de la place en supprimant les imquettes non cellules. En effet, celles-ci sont laissées pour le moment afin de vérifier la qualité de l'algorithme, mais il est prévu, à l'aide d'un paramètre, de choisir leur suppression. Comme une lame compte beaucoup de vide, détectée par les germes, cela économiserait au bas mot la moitié de la place que les imquettes de la lame prendrait. Pour l'avenir de cette méthode, cela peut se révéler très utile puisqu'elles n'ont plus d'utilité à partir du moment où elles sont détectées comme non cellule.

Autre point important, le choix du rayon a toujours été laissé à l'utilisateur. Cependant, pour ne pas biaiser le résultat de la classification par type, il est important de prendre au minimum 32 pixels de rayon. En effet, l'algorithme a appris sur des images de cette taille, même pour les petites cellules. De fait, une petite cellule dans un grand vide est synonyme de normalité et l'inverse augmente les chances d'être détecté comme Anormale puisqu'il s'agit d'une caractéristique principale : une cellule anormale est, le plus souvent, grande voir très grande. Il est donc conseillé d'utiliser 32 pixels comme rayon, voir plus pour les anormales très grandes, et pas moins.

Pour finir, ce script ne prend pas en compte d'autres apprentissages que le binaire ou le multiple avec les sept classes citées plus haut. Pour ajouter une classe ou en retirer une, il faudrait modifier des petites choses dans le script pour la reconnaissance et les couleurs. De même, les amas ne sont pas pris en compte actuellement. Selon les résultats prochains, ces ajout feront peut-être l'objet d'une mise à jour de ce petit programme, mais c'est encore trop tôt pour affirmer que cela sera indispensable.

Reconnaissance des amas par rapport aux cellules isolées

Suite à une demande du docteur Hubert Elie, nous allons essayer de reconnaître les amas comme classe à part entière afin de pouvoir les étudier plus en détail. Nous allons tenter de les reconnaître parmi des cellules isolées, grâce aux algorithmes précédents.

Base de données

Qui dit apprentissage dit base de données. Pour ce faire, nous allons utiliser, comme précédemment, une base fabriquée à partir des bases de CytoProcessor et de CellIA. Il faut savoir que seule cette dernière propose des amas. La répartition est organisée comme indiqué sur le tableau ci-dessous :

		Base CellIA	Base CytoPro	Total
ANORMAL		494	1014	1508
	ASCUS	31	-	-
	HSIL	365	-	-
	LSIL	98	-	-
NORMAL		1308	3783	5091
	Dystrophies	75	381	456
	Endocervicale	29	640	669
	Intermédiaire	386	1089	1475
	Parabasale	24	160	184
	Polynucléaire	411	733	1144
	Superficielle	383	780	1163
SOUS-TOTAL		1802	4797	6599
AMAS		691	-	-

Comme nous pouvons le voir, nous avons environ 700 Amas tandis que nous avons environ 7000 cellules isolées. De fait, nous allons faire 10 rotations sur les Amas afin d'avoir une base équilibrée. Nous n'utiliserons pas l'équilibre des types ici puisque nous n'avons pas suffisamment de cellules pour ça.

De là, nous divisons les 7 000 cellules de chaque classe par deux afin d'avoir une base d'apprentissage et une base de test, comme nous avons l'habitude de faire. Enfin, nous lançons l'algorithme sur ces données là. Cependant, cela fait assez peu de données, ce qui fait que les résultats pourraient en être impactés, comme nous allons le voir ci dessous.

Résultat

Une fois l'apprentissage fait, il suffit de lancer l'algorithme sur la base de tests, qui générera un CSV contenant les statistiques. En utilisant le nombre d'epochs ainsi que la taille des images en pixels des essais précédents, nous obtenons les résultats suivant :

	Taille des images en pixel	Nombre d'Epochs	Pourcentage d'amas bien classés (P)	Pourcentage d'amas mal classés (FN)	Pourcentage de cellules seules bien classés (P)	Pourcentage de cellules seules mal classés (FP)	Total Positif (P)
Test N°1	64	25	1.9	98.1	100.0	0.0	49.8
Test N°2	64	25	8.5	91.5	100.0	0.0	53.2

Test N°3	64	50	1.3	98.7	100.0	0.0	49.5
Test N°4	64	100	13.6	86.4	100.0	0.0	55.8
Test N°5	64	10	0.2	99.8	100.0	0.0	49.0
Test N°6	64	20	0.6	99.4	100.0	0.0	49.2
Test N°8	32	25	13.1	86.9	100.0	0.0	55.5
Test N°9	16	25	2.1	97.9	100.0	0.0	50.0
Test N°11	128	25	4.1	95.9	100.0	0.0	50.9
Test N°12	32	10	9.1	90.9	100.0	0.0	53.5
Test N°14	32	5	5.6	94.4	100.0	0.0	51.7
Test N°19	128	5	13.8	86.2	99.9	0.1	55.9
Test N°20	64	1	6.8	93.2	100.0	0.0	52.3
Test N°21	32	1	2.1	97.9	100.0	0.0	49.9
Test n°27	64	75	1.8	98.2	100.0	0.0	49.8
Test n°34	32	15	6.2	93.4	100.0	0.0	52.0
Test n°35	32	30	13.7	86.3	99.9	0.1	55.8
Test n°36	32	50	8.1	91.9	100.0	0.0	53.0
Test n°37	32	40	20.1	79.9	99.9	0.1	59.1

Comme nous pouvons le constater, les résultats ne sont pas exploitables. En effet, pour une raison encore obscure à l'heure actuelle, la reconnaissance n'excède pas 20% sur les amas, alors que les cellules isolées ne posent jamais aucun souci. De plus, les images sont clairement différentes, ce qui ne donne encore moins d'explications sur ces résultats.

Il faudrait creuser en profondeur sur les raisons de tels résultats, inexplicable à l'heure actuelle, mise à part le manque de données évident. Cependant, selon les spécialistes, il paraît compromis de pouvoir traiter les amas. En effet, si les reconnaître peut effectivement se faire, il est souvent impossible de les analyser. De fait, cet apprentissage n'a pas tant d'importance que ça, si ce n'est qu'enlever des choses inexploitables, ce que l'algorithme précédent fait.

Par contre, cela serait intéressant pour les médecins de savoir qu'il existe des amas, afin qu'ils puissent vérifier si ceux-ci peuvent potentiellement cacher de cellules cancéreuses. Mais puisque nous reconnaissons les anormales, l'utilité reste limitée à un cas qui a été révélé être anormale seulement dans un amas, ce qui peut correspondre à une marge d'erreur raisonnable. Aussi, il faut savoir qu'il n'est pas permis de poser un diagnostic sur les amas.

De fait, cette partie ne semble pas la plus importante au sein du projet. Si nous le balayons provisoirement, telle une feuille morte, ce point sera revu par la suite, afin de comprendre et améliorer cet apprentissage, notamment en injectant quelques caractéristiques et une meilleure maîtrise des données sur les amas. Pour le moment, nous considérons les amas sans distinction possible de cellules comme des choses qu'on ne traitera pas (en non cellule) comme cela a été suggéré par les médecins eux-même afin de continuer à en ressortir les anormales.

Réalité terrain de la reconnaissance de cellules anormales sur une tuile

Afin de conclure réellement sur l'efficacité des algorithmes présentés, nous les avons soumis à un spécialiste en cytologie. En effet, même si les tests étaient faits sur des images dont nous connaissons le type, il était important de tester sur d'autres tuiles, sélectionnés au préalable, afin d'évaluer le résultat et le taux d'erreurs.

Deux tests ont été effectués à ce jour. Puisque l'algorithme qui reconnaît les types n'est pas assez performant, nous avons choisi d'utiliser d'abord celui de reconnaissance de cellules, à taux de réussite de 95%, et ensuite celui de reconnaissances des anormales, à 90% de taux de réussite. Il a été choisi pour un premier lancement 45 tuiles provenant de deux lames différentes (une normale et une anormale). Pour le second test, 30 tuiles (dont 5 venant d'une lame précédente) ont été choisies, ainsi que 25 autres venant d'une coloration différente.

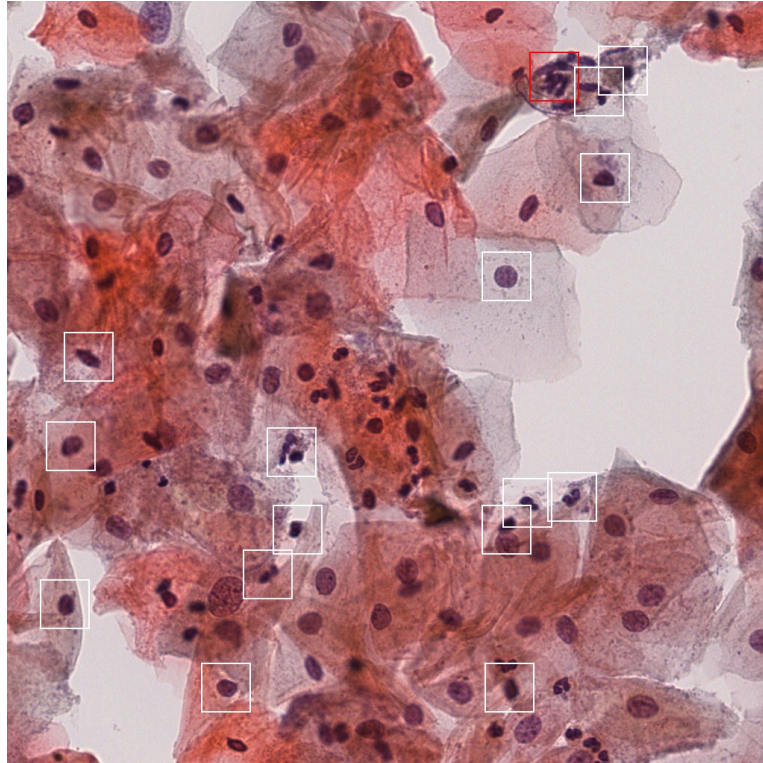
Après le passage de l'algorithme, comme l'explique les parties précédentes sur le petit programme pour détecter les cellules sur une tuile, les cellules normales seront indiquées en blanc tandis que les cellules anormales seront indiquées en rouge. Par la suite, l'expert en cytologie va indiquer sur l'images les erreurs : un point blanc lorsqu'il s'agit d'une erreur sur l'algorithme de détection des cellules (soit un oubli de cellule, soit une cellule qui n'en est pas une) et un point rouge lorsqu'il s'agit d'une erreur sur l'algorithme de détection des anormales (soit un oubli, soit une erreur de classification). Il est à noter qu'une cellule détectée anormale qui n'est, en réalité, pas une cellule, sera compté comme une erreur de détection de cellules. Pour illustrer, voici deux images avant et après correction.



Afin de pouvoir tirer des conclusions, tous ces résultats ont été rangé dans un tableau récapitulatif afin d'évaluer la qualité de la reconnaissance.

Résultats

Avant de présenter le tableau, plusieurs tuiles ont été évincés de l'affaire. En effet, comme dit précédemment, une coloration différente avait été donnée, basée sur le rouge. Nous pouvons en voir un exemple ci-dessous.



Comme nous pouvons nous en douter, l'algorithme de reconnaissance des cellules est inefficace puisque l'entraînement comprenait des cellules colorées avec la coloration Papanicolaou, soit le cytoplasme en turquoise ou rose selon le type de cellules et le noyau en violet. Sur une coloration rouge comme celle-ci, très peu utilisée chez nous, il est normal de voir l'algorithme ne pas fonctionner correctement. Il s'agissait d'une demande de l'expert malgré tout, c'est pourquoi nous le signalons, mais pour pouvoir traiter ce genre de cas, il faudrait prévoir la base adéquate (comme pour les autres sortes de cellules ou coloration). Cependant, vu que cette coloration est rare (et non utilisée par chez nous), nous n'allons pas prendre de mesure vis à vis de ça, simplement constater le résultat.

Pour en revenir à ce qui nous intéresse, voici les tableaux récapitulatifs des tests. Pour comparer, nous avons séparé les erreurs type "oubli" des erreurs de classifications.

Première fournée de tuiles

	Donné par l'Algorithme			Corrigé par un expert (Michel Lécluse + Grégoire Olivier)					
	Nombre de Germes	Nombre de cellules	Nombre d'Anormales	Erreurs Cellules	Oublie Cellules	Erreurs Anormales	Oubli Anormales	Nombre de cellules	Nombre d'Anormales
Total	3443	1384	339	85	35	271	1	1334	29
Moyenne	76,51	30,76	7,53	1,89	0,78	6,02	0,02	29,64	0,64
Ecart Type	41,97	17,59	6,98	2,85	2,62	6,09	0,15	17,23	0,85
Variance	1761,76	309,25	48,78	8,14	6,84	37,13	0,02	296,94	0,72

Pourcentage de cellules trouvées fausses	6,141618497
Pourcentage d'anormales trouvées fausses	79,94100295
Pourcentage d'oublis de cellules	2,623688156
Pourcentage d'oublis d'anormales	3,448275862

Deuxième fournée de tuiles

	Donné par l'Algorithme			Corrigé par un expert (Michel Lécluse + Grégoire Olivier)					
	Nombre de Germes	Nombre de cellules	Nombre d'Anormales	Erreurs Cellules	Oublie Cellules	Erreurs Anormales	Oubli Anormales	Nombre de cellules	Nombre d'Anormales
Total	2204	1018	319	47	74	245	7	1045	56
Moyenne	73,47	33,93	10,63	1,57	2,47	8,17	0,23	34,83	1,87
Ecart Type	28,76	13,22	6,86	2,04	4,79	6,76	0,56	14,83	2,96
Variance	827,38	174,86	47,03	4,18	22,98	45,74	0,31	220,01	8,78

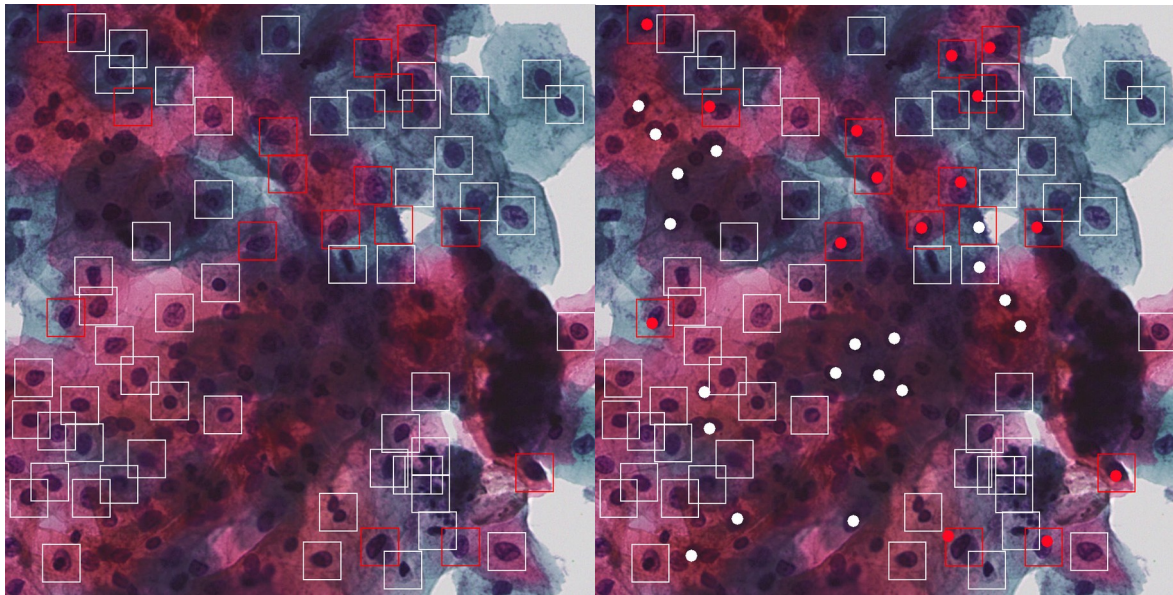
Pourcentage de cellules trouvées fausses	4,616895874
Pourcentage d'anormales trouvées fausses	76,80250784
Pourcentage d'oublis de cellules	7,081339713
Pourcentage d'oublis d'anormales	12,5

Les tableaux complets pour chaque tuile est disponible sur le lien suivant :

<https://drive.google.com/open?id=18Zkfsq30SDImYBiPNABvSAQ7Xo1o3oC9UXiZog2vgvw>

Comme nous pouvons le constater, l'algorithme a tendance à récupérer plus de choses qu'il n'y a déjà, avec une tendance à doubler certaines cellules. Cependant, il est appréciable de voir, dans le cas de la reconnaissance de cellules, qu'il ne fait quasiment aucun oubli. Si le modèle choisi (soit le Test n°9) a un taux de 99% des cellules reconnues, les chiffres sur le terrain le confirme. Le pourcentage d'erreur reste aussi dans la borne acceptable (sachant que beaucoup d'entre elles sont des doublons). Dans les erreurs communes, il y a aussi pas mal d'endroits sombres, où la frontière entre cellules et mucus

est très fine. Comme ces cellules ne peuvent être exploitées, à part pour signaler leur présence, nous pouvons confirmer que la reconnaissance des cellules est efficace. Cependant, celle-ci a un point faible, que nous allons illustrer ci-dessous.



En effet, elle a du mal sur les cellules claires (qui sont rares et généralement pas anormales, puisque le taux de chromatine n'est pas élevé) et sur les cellules roses. Sur ces dernières, le noyau violet est légèrement recouvert de cytoplasme rose, ce qui inhibe le contraste et rend donc la reconnaissance plus compliquée. Sur des images denses, la reconnaissance n'est donc plus suffisante. Pour l'améliorer, il faudrait ajouter des cellules de ce type, afin que l'algorithme les reconnaisse mieux. S'il y en a dans la base de données, ce n'est pas suffisant en terme de diversité. La partie suivante montrera ces ajouts.

Mais avant cela, si nous avons conclu sur la reconnaissance de cellules, qui est plus que satisfaisante, revenons sur la reconnaissance des anormales. Comme attendu, beaucoup plus de cellules anormales qu'il y en a sont reconnues (environ 80%, comme indiqué, n'en sont pas). Les oublis, rares, confirment les statistiques de l'apprentissage, qui donnait 90% de reconnaissances des anormales. Si cela n'est pas suffisant et demande une base de données plus fournies avec les améliorations déjà évoquées, nous pouvons tout de même remarquer que nous avons bien trop de cellules "parasites" dans les anormales.

Cela est principalement dû à l'algorithme précédent. En effet, la base de données pour reconnaître les cellules anormales est réduite, mais surtout extrêmement propre. Nous y retrouvons principalement que des belles cellules, que ce soit anormale ou normale. Mais en réalité terrain, des cellules peuvent baver ou avoir une forme plus éthérée. Dans notre cas, la reconnaissance de cellules donnent aussi des amas ou d'autres choses qui ne sont pas des cellules, à analyser (comme du mucus). De fait, comme ces objets ne sont clairement pas des cellules, et ont une forme irrégulière, elles sont classées comme Anormale. Si cela est logique et inévitable, c'est tout de même un vrai problème pour l'expert, qui sera encombré de choses inutiles. Il faut donc limiter ces arrivées.

Afin de pallier au problème, une nouvelle classe verra le jour avec un apprentissage en plus : vraies ou fausses anormales. Le but ici est de récupérer les cellules corrigées par l'expert et de les mettre dans deux classes différentes. En faisant un apprentissage dessus, nous pourrions en sortir un modèle qui pourra, nous espérons, éliminer les débris et ainsi limiter les objets en trop. Cette correction sera utile, puisque nous ne pouvons éviter toutes les saletés possibles, en particulier les amas.

Enfin, il reste toujours le souci du rayon, qui devrait pouvoir s'adapter autour des cellules. Cette idée est évoquée depuis le départ, et permettrait de ne couper aucune cellule (même si celles-ci sont tout de même parfaitement reconnues la plupart du temps). Cependant, le code doit déjà être mis en place dans une version de l'algorithme des germes de Pierre et pourrait bien aider à la reconnaissance à la fois des cellules, mais surtout des cas anormaux.

Corrections apportées

Maintenant que nous avons vu les points faibles, nous allons pouvoir y pallier dans cette partie.

Tout d'abord, revenons sur la base de reconnaissance entre les cellules et les débris. Comme nous avons pu le constater, si l'algorithme reste excellent, il a quelques points que nous allons essayer de corriger. Pour ce faire, vu que le plus gros problème vient des cellules roses, nous allons en ajouter dans la base de données.

Mais avant cela, la précédente base a été corrigée, via un expert, afin d'enlever les erreurs de classement. En effet, cette partie avait été faite manuellement par moi même, mais quelques cellules étaient passées à la trappe, principalement des polynucléaires qui s'entassaient les uns sur les autres et quelques cas douteux. Cette vérification permet de croire en la base, bien plus qu'avant.

En ce qui concerne l'ajout de cellules, une lame très rose a été donnée, et, grâce au script déjà décrit auparavant, nous avons pu récupérer sur la moitié de la lame, environ 50 000 cellules rosées et 200 000 non cellules. Le tout a été ajouté à la base avant de lancer l'apprentissage et la batterie de tests que nous avons résumé dans le tableau ci-dessous.

	Taille des images en pixel	Nombre d'Epochs	Pourcentage de cellules bien classés (P)	Pourcentage de cellules mal classés (FN)	Pourcentage de non cellules bien classés (P)	Pourcentage de non cellules mal classés (FP)	Total Positif (P)
Test N°1	64	25	97.9	2.1	94.2	5.8	95.9
Test N°2	64	25	82.5	17.5	99.4	0.6	91.8
Test N°3	64	50	87.6	12.4	98.8	1.2	93.8
Test N°5	64	10	94.0	6.0	98.1	1.9	96.2
Test N°6	64	20	97.4	2.6	96.4	3.6	96.9

Test N°7	64	15	93.8	6.2	97.9	2.1	96.1
Test N°8	32	25	88.5	11.5	98.9	1.1	94.2
Test N°9	16	25	98.9	1.1	94.4	5.6	96.4
Test N°10	16	10	85.4	14.6	99.0	1.0	92.9
Test N°12	32	10	88.1	11.9	99.1	0.9	94.2
Test N°13	64	5	95.8	4.2	97.1	2.9	96.5
Test N°14	32	5	81.0	19.0	99.4	0.6	91.2
Test N°15	16	5	97.4	2.6	96.2	3.8	96.7
Test N°16	16	8	94.8	5.2	97.5	2.5	96.3
Test N°17	32	8	97.7	2.3	96.9	3.1	97.3
Test N°18	64	8	97.1	2.9	95.5	4.5	96.2
Test N°19	128	5	88.9	11.1	98.4	1.6	94.1
Test N°20	64	1	93.9	6.1	97.3	2.7	95.8
Test N°23	16	3	94.3	5.7	96.8	3.2	95.7
Test N°24	32	3	90.8	9.2	98.4	1.6	95.0
Test N°25	64	3	96.5	3.5	96.8	3.2	96.7

	Taille des images en pixel	Nombre d'Epochs	Pourcentage de cellules bien classés (P)	Pourcentage de cellules mal classés (FN)	Pourcentage de non cellules bien classés (P)	Pourcentage de non cellules mal classés (FP)	Total Positif (P)
Test N°9 BIS	16	25	92.8	7.2	98.5	1.5	96.0

Le tableau complet est disponible sur le lien suivant : https://drive.google.com/open?id=1p19TrZE9pjcNe9dJAq9Kgn_Owj3CkIOSffCo5gGITH0

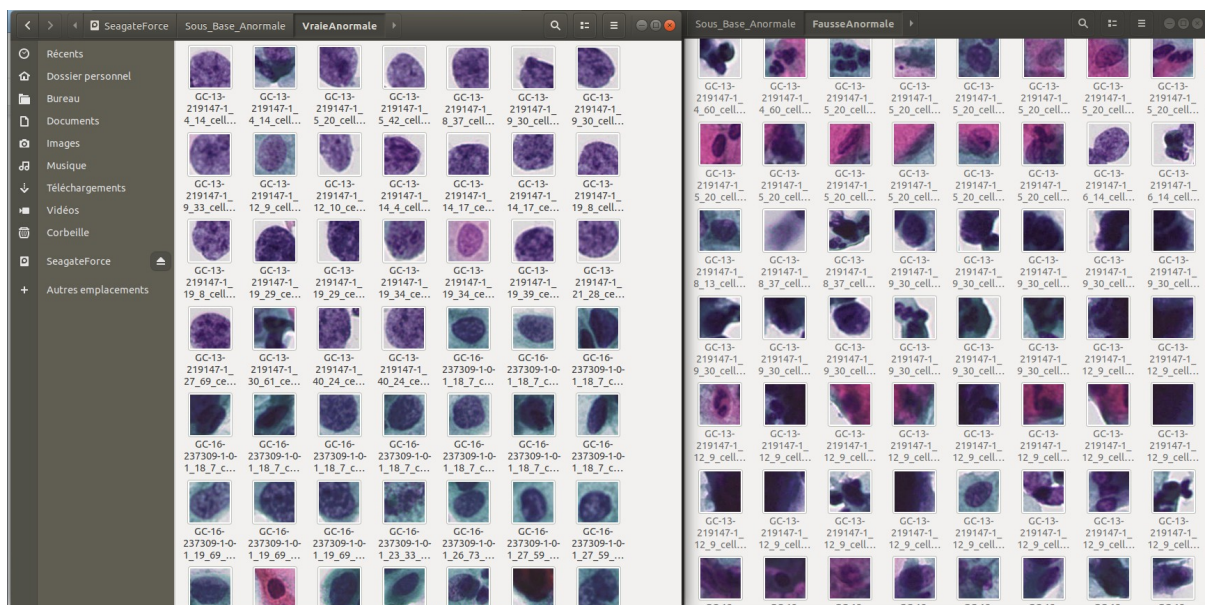
Comme nous pouvons le constater, nous avons une bien meilleure reconnaissance des cellules en général, atteignant 2% de plus qu'avant, soit jusqu'à 97,3% en moyenne. Pour comparer, et nous assurer qu'il ne s'agit pas seulement de la correction de la base, nous avons comparé avec seulement la base corrigée, qui ne gagne que 0,5% en plus. Cependant, plus amples tests seraient préférables. Ici, cela n'a pas pu être réalisé par manque de temps.

Malgré tout, nous posons un petit bémol. Si l'algorithme est plus équilibré par l'ajout dans la base, nous perdons des points pour reconnaître les cellules. Cela peut se remarquer, et même les cellules roses, pourtant mieux en masse, ne sont pas mieux reconnues, au contraire. Nous l'expliquons d'abord par le fait qu'il y a un déséquilibre non voulu entre cellules et non cellules dans la base, avec beaucoup de cytoplasmes roses, ce qui n'aide en rien à reconnaître les cellules roses lorsque la couleur du noyau est proche du cytoplasme. Mais aussi, lors des ajouts, nous ne traitons trop peu des cas aussi compliqués que ceux que je viens d'expliquer, et que nous avons illustré un peu plus tôt dans ce rapport. Pour parfaire la base, il faudrait réellement équilibrer la base entre tous les cas, ce qui n'est

pas simple en pratique sans outils adaptés. Dans tous les cas, pour grappiller les derniers points de pourcentages, il faudra sans doute en venir là, ce qui sera, nous l'espérons, bien plus simple avec les outils en cours de développement.

Enfin, revenons sur les anormales. Comme nous avons pu le constater, 80% des anormales détectées sont des cellules normales souvent étalées ou des débris. De fait, l'idée était venu de faire un traitement là dessus afin de faire ressortir les vraies Anormales. Pour ce faire, il s'agit de rassembler une base de données pour faire un apprentissage sur les vraies Anormales et les fausses Anormales. Pour la réaliser, nous avons utilisé le résultat des tuiles testées dans la réalité terrain, ce qui fait 88 vraies anormales et 577 fausses anormales. Même s'il est possible d'utiliser la rotation, le nombre reste faible pour garantir le bon fonctionnement de l'algorithme. C'est pourquoi, comme pour la construction de la base sur les cellules et non cellules, un petit script a été créé.

Le but du script est assez simple. Il reprend tout simplement les algorithmes précédents. Tout d'abord, il faut lancer l'algorithme de reconnaissance sur la tuile, afin de créer le fichier CSV comprenant la classification de chaque cellule, comme nous avons pu le décrire dans les parties précédentes. Ensuite, le petit script prend son essor : il va lire le dit fichier et afficher seulement les cellules dites anormales. L'utilisateur peut alors indiquer lorsqu'une cellule est réellement anormale, en mettant un trait à l'intérieur du carré autour de la cellule, comme cela a été fait pour construire la base cellule/non cellules. Enfin, le script va en extraire les imagerie et les ranger dans les deux catégories, donnant deux dossiers comme ci-dessous.



Bien entendu, la provenance de chaque cellule est indiquée dans le nom de l'imagerie afin de la retrouver au besoin. Il ne manque qu'à faire l'apprentissage sur cette petite base, en y ajoutant des rotations pour l'agrandir. Cependant, par manque de temps, cette partie s'arrêtera ici. Nous espérons qu'elle sera réalisée avec le spécialiste Youssef Chahir, afin de pouvoir grandement améliorer la reconnaissance des Anormales, du moins

avoir bien plus de choses pertinentes qu'à l'heure actuelle, avec 80% d'objets qui n'ont rien à voir (et facilement évitable sur la différence que nous pouvons constater sur la présente image).

Et c'est ici que les corrections s'arrêtent. Nous pouvons donc conclure que la reconnaissance des cellules aux débris est suffisamment efficace et, même si nous pensons encore pouvoir l'améliorer, pourra être intégré comme prétraitement, afin de cibler le mieux possible les cellules. Couplée avec le rayon qu'elle pourrait avoir, nous aurons un outil très puissant pour capter les noyaux, ce qui est le but de la manoeuvre. Quant à la reconnaissance des anormales, si préciser le type semble compliqué, la reconnaissance d'anormales, avec les améliorations prévues, pourrait devenir très efficace et permettre aussi un prétraitement, afin de limiter des calculs avec cet algorithme puissant. Le deep learning apporte bien ses preuves, même en réalité terrain et les corrections effectuées pourront, nous espérons, parfaire les bons résultats et ainsi pouvoir aider grandement les médecins et éviter bien des problèmes aux patients.

Pistes à explorer

Après tous ces résultats concluants, nous n'avons plus à prouver l'efficacité du deep learning sur la reconnaissance des cellules d'abord, et de leur anormalité ensuite. De fait, le projet se poursuivra sous cette même étoile en différents points pour à la fois améliorer l'existant, mais aussi tenter des nouvelles techniques afin de pouvoir trouver de meilleurs résultats, notamment dans la reconnaissance des types. Nous allons donc énoncer les différentes choses à améliorer sur tout le travail effectué jusque là :

- **Continuer les tests terrain pour obtenir une base de données suffisante pour enlever les "fausses" anormales** : comme expliqué sur la réalité terrain, afin d'enlever le maximum de cellules faussement anormales (car abîmée ou débris), nous allons ajouter une étape d'apprentissage sur les corrections de la réalité terrain. Il faudrait donc augmenter le nombre de cellules de cette base et faire un apprentissage pour en voir les résultats.
- **Obtenir une base de données de cellules par leur type et refaire l'apprentissage en y insufflant des caractéristiques** : à l'aide de M. Michel Lecluse et d'étudiants sur un projet d'outils, nous pourrions obtenir une base de types de cellules conséquente afin d'améliorer la reconnaissance des différents types de cellules. Aussi, vu la proximité de certaines classes, il faudrait ajouter quelques caractéristiques, notamment la taille, la forme et la texture. L'environnement serait aussi une bonne piste à explorer.

- **Apporter des améliorations sur l'algorithme de deep learning** : que ce soit pour les types de cellules ou bien la séparation des cellules par rapport aux non cellules, l'algorithme n'a pas été personnalisé pour ces cas de figure, alors que les cellules ont des caractéristiques bien précises. Le but serait de proposer des opérations mathématiques qui mettent en valeur ces caractéristiques afin d'améliorer la reconnaissance, notamment à l'aide de Youssef Chahir, spécialiste dans le domaine.
- **Rassembler les étapes précédentes sur une lame entière** : pour des questions pratiques, nous ne travaillons que sur une tuile, ce qui peut couper des cellules. Aussi, le temps de calcul est actuellement trop long pour passer autant de tuiles, bien que facilement optimisable par un spécialiste. Les étapes précédentes devront s'adapter à une lame entière, via les logiciels et algorithmes faits auparavant.

Il n'y a pas d'ordre spécifique sur ces points. Si les améliorations sont en bonne voie d'être travaillées, les scripts permettant de faire la détection ne demandent qu'à être optimisés et intégrés aux logiciels existants, comme prétraitement pour la reconnaissance cellules/non cellules. Cependant, cela sera aux suivants et aux autres membres de l'équipe de continuer sur ces pistes, et d'autres peut-être si des idées surviennent.

Conclusion

Tout au long de ce rapport, nous avons pu voir que les travaux effectués depuis le 19 Mars 2018, leurs avancements et les prochaines prévisions. Si les premiers résultats étaient très positifs pour reconnaître les cellules (jusqu'à plus de 95% de reconnaissance entre les cellules et les non cellules), les autres travaux accentuent la pertinence du deep learning dans le cadre de la reconnaissance des cellules cancéreuses.

Si d'autres travaux ne donneront pas de suites (détection plus générale sans imagerie, détection sur les amas) ou bien donnent trop d'erreurs (reconnaissance de tous les types), l'avancée générale montre bien que malgré les multiples possibilités qui ne s'équivalent pas, il existe toujours une solution, parfois bateau, qui ne demande qu'à être trouvée et mise en pratique.

Aussi, les algorithmes sont encore perfectibles pour mieux coller à l'apparence des cellules. Mais nous savons d'ors et déjà que nous rendrons service à de nombreuses personnes et ainsi lutter efficacement contre le cancer. Que ce soit la reconnaissance des images portant des cellules, ou bien de leur emplacement sur un morceau de lame de microscope, qui demande qu'à être évaluée, ou encore l'isolation des cellules anormales, nous avons que des résultats très positifs, atteignant la haute sphère des pourcentages de réussite. En continuant à creuser dans cette voie, une solution efficace est en mesure d'être trouvée, à l'aide des cytopathologistes. A terme, nous espérons que cet optimisme se réalise et concrétise un logiciel efficace à la détection des cancers.

Le projet se termine sur de bonnes choses. Cependant, il reste des choses à explorer et à intégrer, afin d'arriver à un résultat suffisamment satisfaisant et un logiciel compétent pour le diagnostic des cancers.