

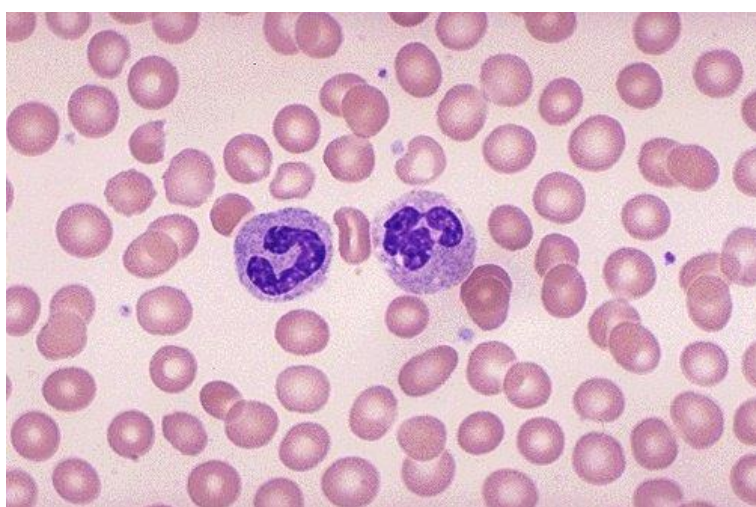


DataScientest • com

Rapport Technique d'évaluation

Projet BLOOD CELLS

Reconnaissance des cellules sanguines en Computer Vision



Promotion : JUIN 2021 Data Scientist, formation continue

Participants :

Laure BERNARD

OBJECTIFS	2
DONNÉES	2
Base Mendeley.....	2
Base Kaggle Acute Promyelocytic Leukemia	5
PROJET	6
Identification de la problématique	6
Description des travaux réalisés	6
1. Premiers modèles	6
2. Modèles Deep Learning	8
3. Analyse	9
4. Segmentation	12
5. Ajout de 2 classes : Smudge et Blast	14
Difficultés rencontrées lors du projet.....	15
BILAN & SUITE DU PROJET	16
ANNEXES.....	17

Objectifs

L'objectif principal était de pouvoir identifier certains types de cellules sanguines à partir de leur image.

Les cellules présentes dans le sang sont divisées en 3 catégories : les globules rouges, les globules blancs et les plaquettes. La numération globulaire (% de globules rouges, blancs et plaquettes) dans le sang permet de révéler des problèmes de santé : anémie, infection, hémorragie... La reconnaissance des différents types de globules blancs (leucocytes) va permettre d'identifier la présence de cellules dans le sang qui n'y sont normalement pas présents, mais présents uniquement dans la moëlle osseuse.

La présence de ces types de cellules dans le sang en quantité importante peut être le signe de problèmes de santé, tel que des leucémies.

L'objectif était donc de pouvoir identifier les différents types de globules blancs (leucocytes), ce qui permet, lors d'une analyse d'échantillons sanguins d'identifier des anomalies pouvant révéler d'une maladie.

J'ai pris contact et rencontré une hématologue, intervenant dans un laboratoire biologique, afin de bien identifier les différents types de leucocytes et de comprendre le fonctionnement d'une analyse sanguine. Cela m'a permis de définir les types de cellules qu'il est indispensable d'identifier dans un échantillon sanguin.

Actuellement, un système automatisé permet de différencier une quinzaine de type de cellules. La technique n'utilise pas d'intelligence artificielle mais des algorithmes de segmentation afin d'obtenir différentes caractéristiques : diamètre/périmètre de la cellule, diamètre/périmètre du noyau, couleur, textures....

Données

Deux jeux de données ont été utilisés

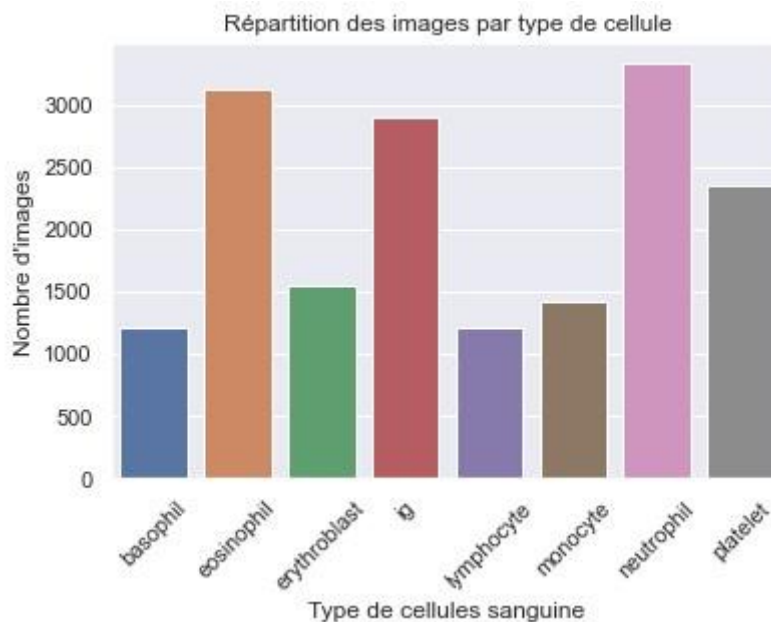
Base Mendeley

Le dataset est disponible sur le site de partage de ressource Mendeley: « A dataset for microscopic peripheral blood cell images for development of automatic recognition systems »

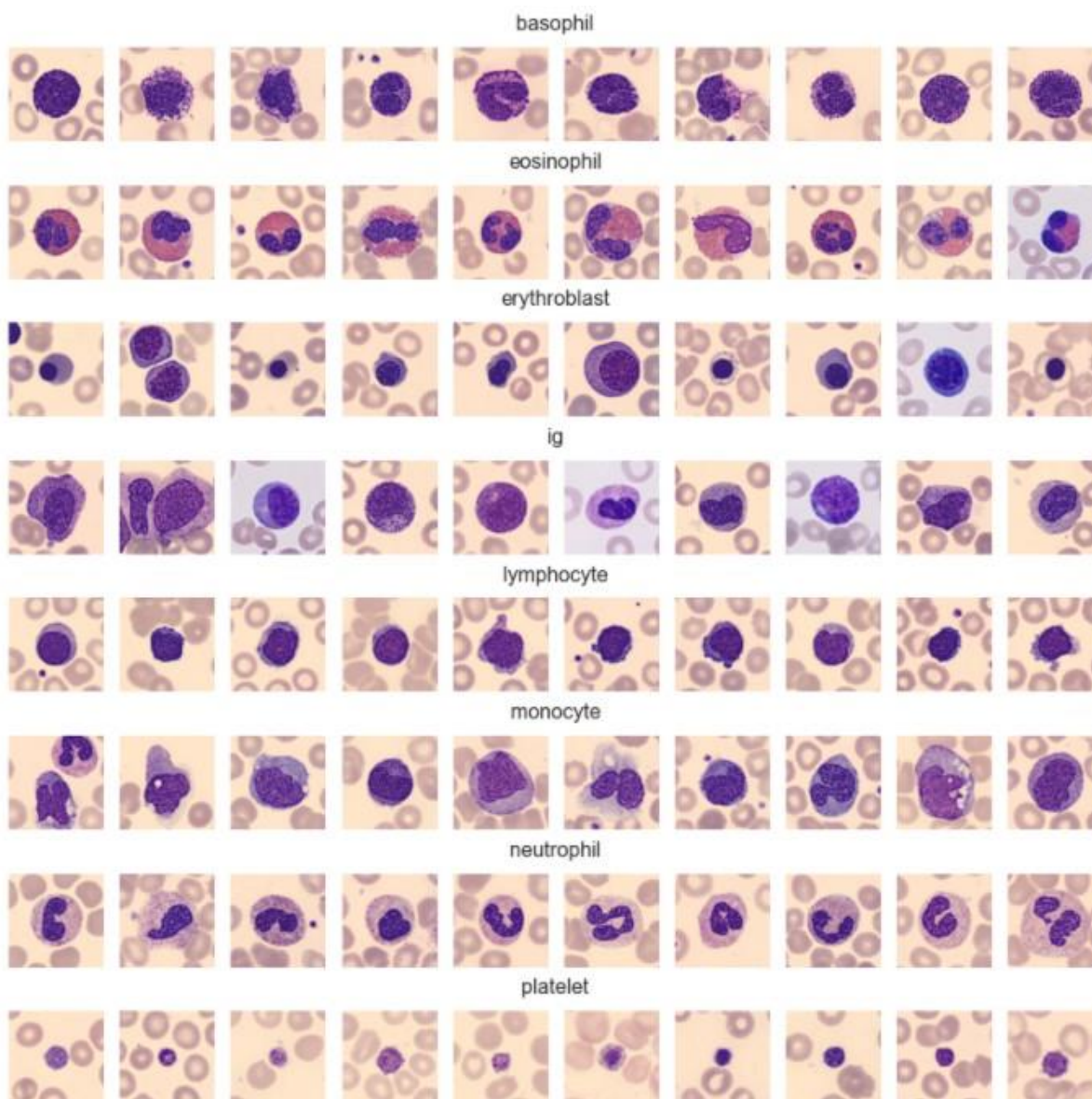
<https://data.mendeley.com/datasets/snkd93bnjr/1>

Elle contient 17092 images de cellules sanguines de 8 types différents :

- Types de leucocytes (globules blancs) : basophil, eosinophil, erythroblast, ig, lymphocyte, monocyte, neutrophil. Les IG (immatures granulocytes) regroupent : métamyelocytes, myelocytes et promyelocytes
- Plaquette : platelet



Visualisation des différents types

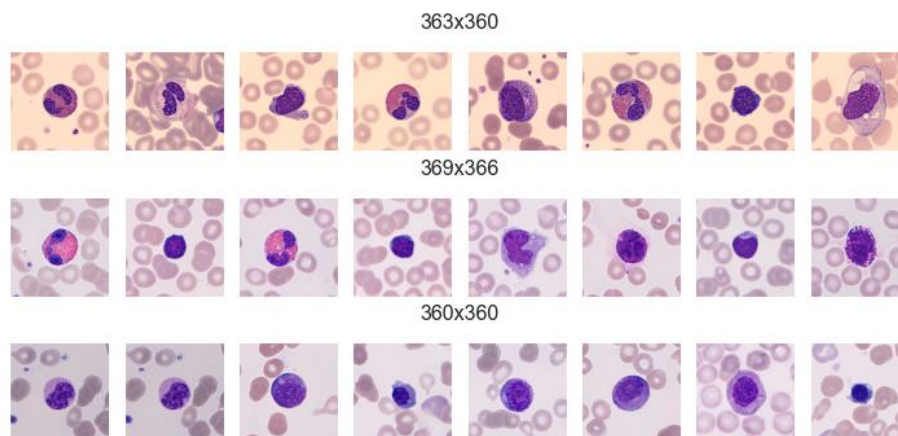


Les plaquettes (platelets) sont facilement identifiables car plus petits. Les eosinophils semblent avoir un noyau d'une autre couleur. Les autres types sont plus difficilement différenciables.

Seules quelques images ne sont pas au format 363x360 :

363x360	16639
369x366	250
360x360	198
361x360	2
360x362	1
360x359	1
360x361	1

J'ai pu identifier que les images au format 369x366 et 360x360 présentent une luminosité différente.



Les images avec une luminosité différente se retrouve dans 7 types de cellules :

ig	147
erythroblast	51
basophil	50
eosinophil	50
lymphocyte	50
monocyte	50
neutrophil	50

Les images à faible luminosité étant réparti dans les différents types de cellules et étant en petite quantité, nous n'identifions pas ici de biais dans les données.

Seulement 17 doublons ont été identifiés, parmi eux une anomalie a été relevé : la même image a été classée dans 2 types différents. Ceci révèle la difficulté, même pour le biologiste de classer les différents types de cellules.

Sur toutes les images, les cellules sont centrées. Les images peuvent être rognées au format 256*256 à partir du centre, aucune perte d'information sur la cellule.

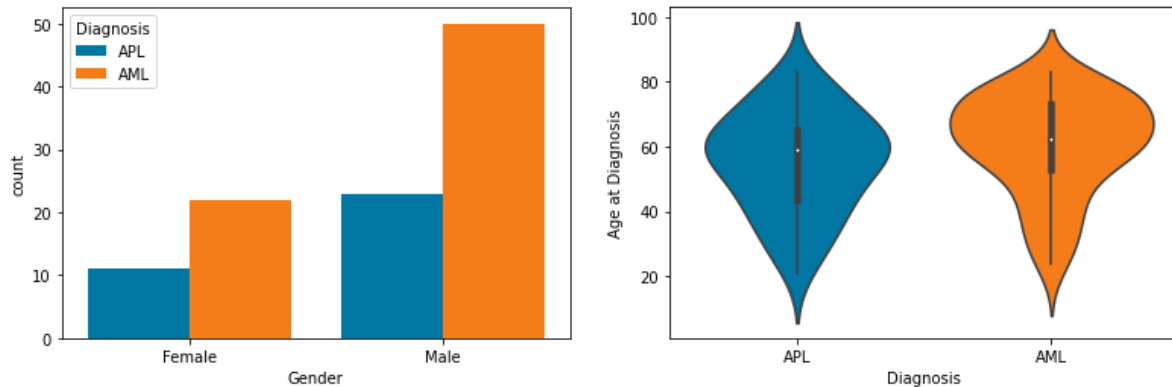
L'intégralité des images a été conservée.

Base Kaggle Acute Promyelocytic Leukemia

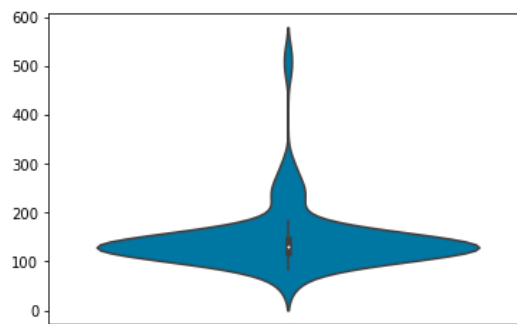
Le dataset est disponible sur le site de partage de ressource Kaggle: « Acute Promyelocytic Leukemia »

<https://www.kaggle.com/eugeneshenderov/acute-promyelocytic-leukemia-apl>

Les images proviennent de 106 patients atteints de leucémie aigue promyélocytaire (APL) ou leucémie aigue myéloïde (AML).



Au total, le dataset contient 15517 images de cellules identifiées : 110 images en moyenne par patient.



Les cellules sont identifiées dans 20 catégories différentes.

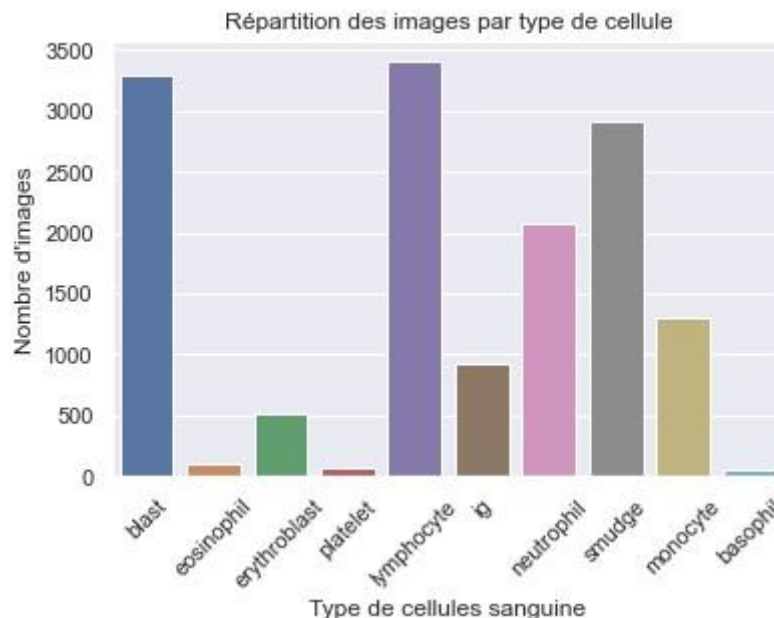
28 doublons ont été identifiés : là encore, chaque image a été classée dans 2 catégories différentes.

Les cellules sont toutes centrées, parfois plusieurs cellules sont agglutinées au centre

Sur les 20 catégories, certaines ont été regroupées, d'autres supprimées car contenant trop peu de données.

Les doublons ont été supprimées ainsi que certaines images buggées (matrice impossible à lire).

La base contient, après nettoyage, 14643 images réparties en 10 catégories : blast, basophil, eosinophil, erythroblast, ig, lymphocyte, monocyte, neutrophil, platelet et smudge.



Certaines catégories comprenant peu de cellules, cette base n'a pas été utilisée seule pour entrainer un modèle mais en complément de la base Mendeley

Projet

Identification de la problématique

Le problème concerne une classification d'image. Je me suis donc orienté rapidement vers des modèles de Deep Learning.

Afin d'évaluer la performance des modèles, la métrique « accuracy » (taux de prédiction correcte) a été utilisée, ainsi que l'analyse des taux de rappel et de précision par type de classe.

Description des travaux réalisés

1. Premiers modèles

Dans un premier temps, j'ai uniquement utilisé la base d'image Mendeley afin d'identifier un modèle permettant de classer les 8 types labellisés dans cette base : basophil, eosinophil, erythroblast, ig, lymphocyte, monocyte, neutrophil et platelet.

Pour servir de baseline, j'ai tout d'abord utilisé 2 modèles de Machine Learning (Random Forest et SVC) sur une base de 4000 images en Noir et Blanc : 500 images par type de cellule.

Nous obtenons déjà un très bon score avec un taux de prédiction correcte (accuracy) de 84% pour le Random Forest et de 80% pour le SVC.

Deux types de cellules donnent des scores nettement plus faibles : 67% pour les IG et 77% pour les monocytes.

Random Forest (entraînement sur 4000 images N&B)

	precision	recall	f1-score	support
basophil	0.79	0.80	0.79	89
eosinophil	0.81	0.83	0.82	100
erythroblast	0.89	0.89	0.89	95
ig	0.69	0.66	0.67	96
lymphocyte	0.87	0.89	0.88	120
monocyte	0.77	0.77	0.77	102
neutrophil	0.89	0.85	0.87	102
platelet	0.99	1.00	0.99	96
accuracy			0.84	800
macro avg	0.84	0.84	0.84	800
weighted avg	0.84	0.84	0.84	800

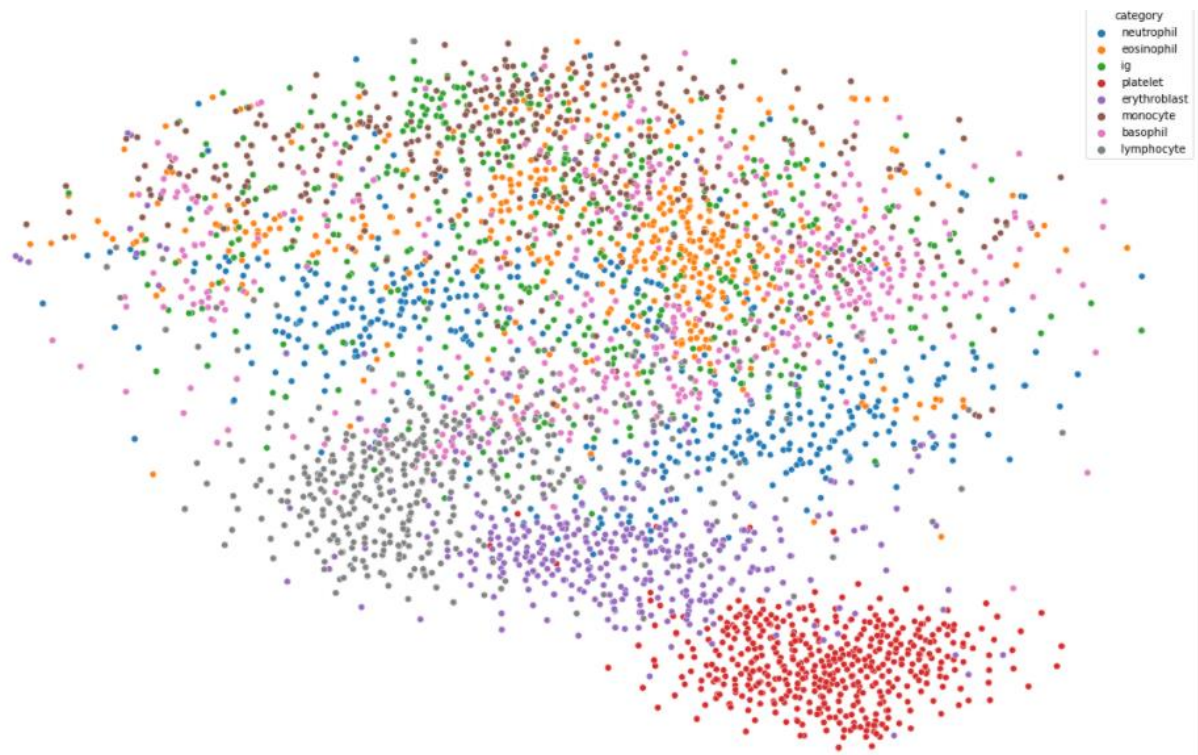
Sur cette même base de 4000 images, un modèle convolutif type LeNet donne une accuracy à 74% avec des images en Noir et Blanc et 87% avec des images en couleurs. Le modèle est cependant en overfitting avec une accuracy de 99% sur les données d'entraînement.

Les cellules de type IG ont là encore du mal à être correctement identifiées : précision à 66% et rappel à 69%.

LeNet (entraînement sur 4000 images en couleurs)

	precision	recall	f1-score	support
basophil	0.94	0.81	0.87	116
eosinophil	0.95	0.93	0.94	99
erythroblast	0.92	0.95	0.94	99
ig	0.66	0.69	0.68	117
lymphocyte	0.96	0.88	0.92	113
monocyte	0.77	0.92	0.84	79
neutrophil	0.84	0.86	0.85	86
platelet	0.98	1.00	0.99	91
accuracy			0.87	800
macro avg	0.88	0.88	0.88	800
weighted avg	0.88	0.87	0.87	800

Afin d'essayer d'identifier des caractéristiques sur les images, j'ai utilisé un modèle de Manifold Learning pour visualiser les données. Trois types de cellules sont facilement identifier sur le graphique (groupés) : erythroblast, platelet et lymphocyte. Ces types de cellules obtiennent effectivement des scores supérieurs à 92% et même 99% pour les platelets.



2. Modèles Deep Learning

J'ai repris le modèle type LeNet précédent mais cette fois sur la totalité des images : 11000 images pour l'entraînement, 2700 pour les tests et 3400 pour la validation.

En augmentant ainsi les données d'entraînement, l'accuracy s'améliore pour passer de 87 à 93%. Le type de cellule IG obtient toujours un score plus faible de 85%.

Afin d'analyser les caractéristiques de l'image permettant au modèle de classifier la cellule, j'utilise l'algorithme de GradCam.

LeNet (entraînement sur 11000 images, base mendeley)

	precision	recall	f1-score	support
basophil	0.81	0.95	0.88	222
eosinophil	0.98	0.96	0.97	627
erythroblast	0.96	0.89	0.92	311
ig	0.86	0.84	0.85	590
lymphocyte	0.86	0.99	0.92	236
monocyte	0.91	0.85	0.88	302
neutrophil	0.95	0.95	0.95	682
platelet	1.00	0.99	0.99	449
accuracy			0.93	3419
macro avg	0.92	0.93	0.92	3419
weighted avg	0.93	0.93	0.93	3419

Transfer Learning

Le principe du Transfer Learning est d'exploiter des modèles complexes et performant de Deep Learning, nécessitant des temps de calcul élevés et des ressources importantes. Ces modèles pré-entraînés sur de grandes quantités de données vont être utilisés comme point de départ de nouveaux modèles pour un apprentissage plus rapide.

Afin d'améliorer encore les performances, le modèle VGG16 en Transfer Learning a été utilisé sur cette même base d'image (11000 en entraînement). Le résultat s'améliore passant à 96%.

Fine-Tuning

Certaines couches du modèle de base vont être « dégelées », c'est-à-dire réentraînées avec la nouvelle base (poids recalculés pendant l'entraînement du modèle).

J'ai testé plusieurs modèles avec une base VGG16 en fine-tuning, en variant le nombre de couche « dégelée » du modèle VGG16 ainsi qu'en changeant d'algorithme d'optimisation de descente de gradient. La meilleure accuracy obtenue est de 98% : 8 dernières couches de VGG16 dégelées, utilisation de Adam puis SGD, couches de 1024, 512 et 256 neurones.

VGG16 en Transfer Learning (11000 images)

	precision	recall	f1-score	support
basophil	0.99	0.99	0.99	222
eosinophil	0.99	0.99	0.99	627
erythroblast	0.98	0.99	0.98	311
ig	0.96	0.97	0.97	590
lymphocyte	0.98	0.97	0.97	236
monocyte	0.98	0.97	0.98	302
neutrophil	0.98	0.98	0.98	682
platelet	1.00	1.00	1.00	449
accuracy			0.98	3419
macro avg	0.98	0.98	0.98	3419
weighted avg	0.98	0.98	0.98	3419

Features extraction (extraction de caractéristiques)

La dernière couche de neurones du modèle de réseau convolutif va être utilisé comme entrée d'un nouveau modèle de Machine Learning.

En réutilisant le modèle précédent en extraction de caractéristiques (Features extraction) avec un modèle SVC, le résultat monte à 98.5% de prédictions correctes.

3. Analyse

Les scores obtenus sont vraiment élevés et peuvent laisser penser à un biais dans les données.

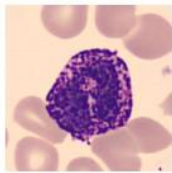
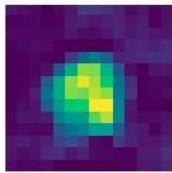
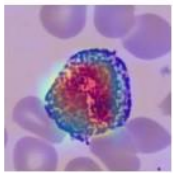
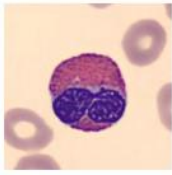
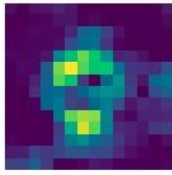
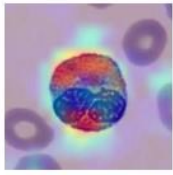
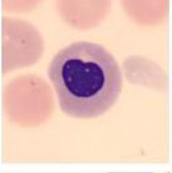
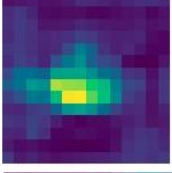
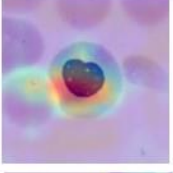
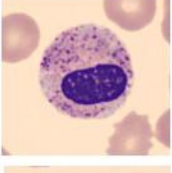
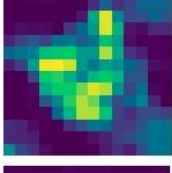
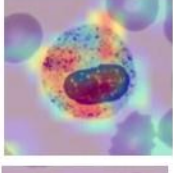
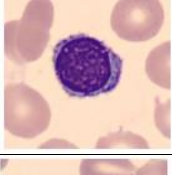
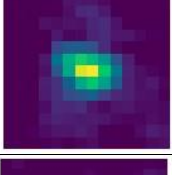
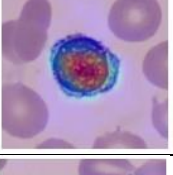
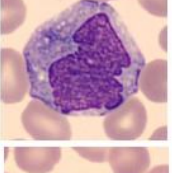
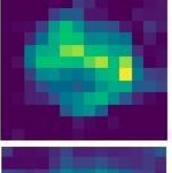
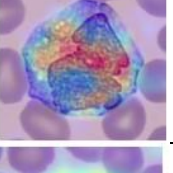
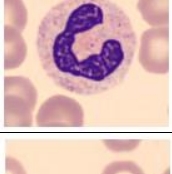
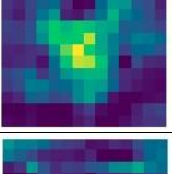
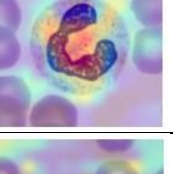

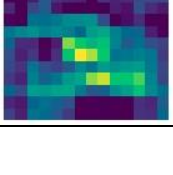

Pour cela, j'ai évalué le modèle des 2727 images provenant de la 2^{ème} base de données (Kaggle) : l'accuracy tombe à 45%.

Le modèle semble donc effectivement avoir un biais pour la classification des cellules dans cette base.

Grad-CAM

La carte d'activation de classe pondérée par le gradient (Grad-CAM) produit une carte thermique qui met en évidence les régions importantes d'une image en utilisant les gradients de la cible (chien, chat) de la couche convolutive finale.

L'utilisation du Grad-CAM n'a pas permis d'identifier de biais.

Basophil			
Eosinophil			
Erythroblast			
IG			
Lymphocyte			
Monocyte			
Neutrophil			
Platelet			

Problématique des IG

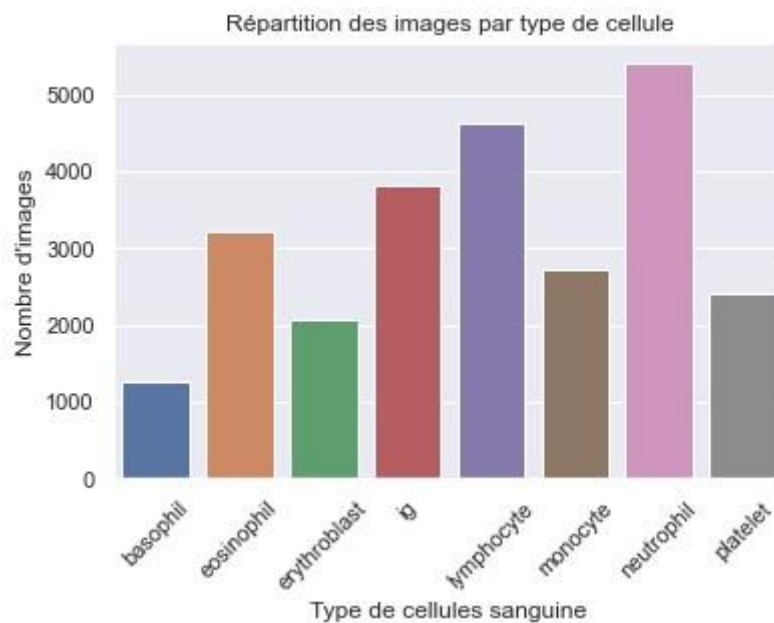
Ayant un score plus faible sur les IG, j'ai essayé de trouver une explication. Sachant que les IG sont des cellules immatures d'autres types (basophil, neutrophil, eosinophil), je me suis demandé si en supprimant la classe IG, ces images seraient alors prédites dans l'une de ces 3 classes. Un modèle entraîné sans IG a finalement prédit ces derniers principalement dans la classe des monocytes. Mon hypothèse de départ n'a donc pas été confirmée.

monocyte	1403
neutrophil	680
basophil	580
erythroblast	151
lymphocyte	53
eosinophil	19
platelet	9

Les 2 classes IG et monocytes semblent donc présenter des caractéristiques communes entre elles et ainsi qu'avec d'autres types de cellules.

Ajout base Kaggle

L'ajout des images de la base Kaggle a permis d'obtenir 25530 images pour les 8 types de cellules.



Avec le modèle type LeNet, utilisé précédemment, entraîné sur cette nouvelle base, l'accuracy obtenue est alors de 88%, avec toujours un score plus faible sur les IG et monocyte. Il était de 93% avec la seule base mendeley.

LeNet sur 2 bases (entraînement sur 20680 images)

	precision	recall	f1-score	support
basophil	0.98	0.84	0.90	118
eosinophil	0.99	0.89	0.94	322
erythroblast	0.87	0.89	0.88	227
ig	0.72	0.83	0.77	413
lymphocyte	0.91	0.92	0.92	470
monocyte	0.79	0.78	0.79	269
neutrophil	0.91	0.89	0.90	507
platelet	0.99	0.94	0.96	227
accuracy			0.88	2553
macro avg	0.90	0.87	0.88	2553
weighted avg	0.88	0.88	0.88	2553

Le résultat plus faible avec les 2 bases révèle une disparité dans les images entre les 2 bases. Les résultats obtenus sont cependant corrects avec un modèle de Deep Learning type LeNet rapide en apprentissage.

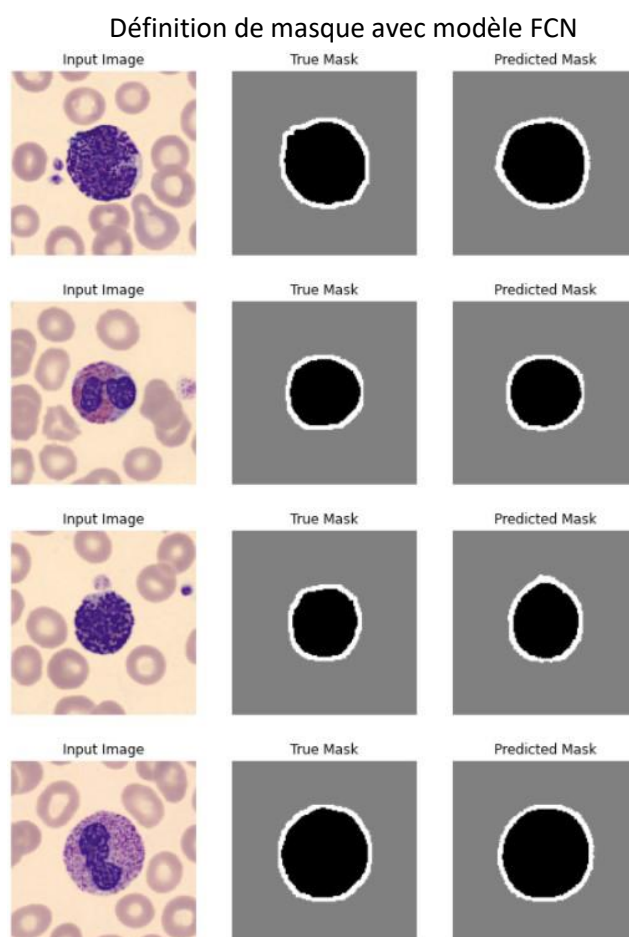
4. Segmentation

Pour essayer d'améliorer les résultats, j'ai pensé modifier les images en supprimant les informations se trouvant autour de la cellule : fond, globules rouges.... En effet, seules les caractéristiques de la cellule (leucocyte ou plaquette) permettent de la classifier : l'environnement n'est pas utile.

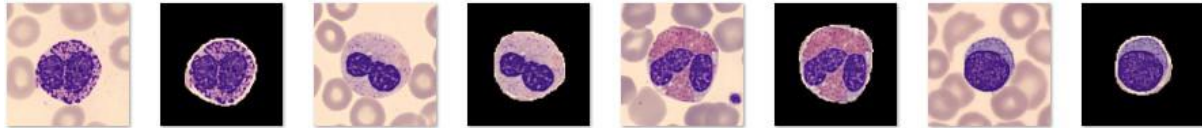
J'ai essayé de définir un algorithme de traitement d'image permettant de détourer la cellule. Cependant, sur un certain nombre d'image, les globules rouges agglutinés autour de la cellule empêchent le fonctionnement correct de l'algorithme.

J'ai ensuite sélectionné 1400 images sans globules rouges agglutinés sur lequel j'ai réussi à détecter les contours via le traitement d'image (OpenCV). A partir de ces images et contours, j'ai utilisé des modèles Deep Learning de segmentation (Unet et FCN), entraînés sur ces 1400 images.

Les 2 modèles ont donné de très bon résultats pour la détection des contours de la cellule.



Même sur des images ayant des globules rouges agglutinés à la cellule, la prédiction avec le modèle FCN donne de bons résultats.



J'ai ainsi recréé une base d'images en supprimant les informations autour de la cellule, remplacées par un fond noir. Cependant, le modèle type LeNet appliqué à cette nouvelle base, permet que très légèrement d'améliorer l'accuracy (1%)

	precision	recall	f1-score	support
basophil	0.89	0.85	0.87	110
eosinophil	0.96	0.96	0.96	305
erythroblast	0.83	0.90	0.86	206
ig	0.80	0.75	0.78	380
lymphocyte	0.91	0.94	0.92	474
monocyte	0.87	0.80	0.83	277
neutrophil	0.90	0.92	0.91	559
platelet	0.98	1.00	0.99	230
accuracy			0.89	2541
macro avg	0.89	0.89	0.89	2541
weighted avg	0.89	0.89	0.89	2541

Le détourage n'améliorant que très peu le résultat, mais ayant trop d'informations autour de la cellule non utiles, j'ai relancé le modèle en recadrant l'image au format 256x256 à partir du centre. La cellule était en effet encore totalement visible.

Les résultats obtenus, sur 2553 images utilisées uniquement pour l'évaluation, sont meilleurs avec une accuracy de 92 % (+4%).

LeNet sur 2 bases avec recadrage 256x256 (entrainement sur 20680 images)

	precision	recall	f1-score	support
basophil	0.99	0.83	0.90	118
eosinophil	0.96	0.95	0.96	322
erythroblast	0.96	0.93	0.94	227
ig	0.78	0.90	0.83	413
lymphocyte	0.94	0.95	0.95	470
monocyte	0.86	0.83	0.84	269
neutrophil	0.96	0.93	0.94	507
platelet	1.00	0.96	0.98	227
accuracy			0.92	2553
macro avg	0.93	0.91	0.92	2553
weighted avg	0.92	0.92	0.92	2553

La classe des IG pose toujours problème avec des confusions avec monocyte, neutrophil, basophil, lymphocyte.

Prédiction	basophil	eosinophil	erythroblast	ig	lymphocyte	monocyte	neutrophil	platelet
Réalité								
basophil	98	3	0	14	1	2	0	0
eosinophil	0	307	0	6	0	1	8	0
erythroblast	0	0	210	7	9	0	1	0
ig	1	2	0	370	7	23	10	0
lymphocyte	0	1	2	12	446	7	2	0
monocyte	0	0	0	43	4	222	0	0
neutrophil	0	7	1	24	2	4	469	0
platelet	0	0	6	0	3	0	0	218

5. Ajout de 2 classes : Smudge et Blast

La base provenant de Kaggle incluait 2 classes supplémentaires que j'ai souhaité ajouter aux modèles. En effet, la classe « Smudge » représente les leucocytes éclatés (lors de la réalisation du frottis) qui ne sont alors pas classifiables dans une autre catégorie. La classe « Blast », quant à elle, représentent des cellules normalement présentes uniquement dans la moëlle épinière mais pouvant se retrouver dans le sang lors de maladie. Il est donc nécessaire de pouvoir les reconnaître afin d'identifier un risque de maladie.

La base obtenue contient 31603 images classifiées en 10 catégories.

En reprenant le modèle type LeNet, l'accuracy obtenu est alors de 85% : la précision sur les blast et le rappel sur les smudge étant nettement plus faible.

LeNet, 10 classes (entrainement 25600 images)

	precision	recall	f1-score	support
basophil	0.81	0.88	0.84	119
blast	0.75	0.92	0.83	327
eosinophil	0.97	0.94	0.95	323
erythroblast	0.93	0.80	0.86	210
ig	0.77	0.65	0.71	381
lymphocyte	0.84	0.92	0.88	453
monocyte	0.73	0.81	0.77	274
neutrophil	0.92	0.91	0.91	573
platelet	0.98	0.97	0.97	242
smudge	0.80	0.66	0.73	259
accuracy			0.85	3161
macro avg	0.85	0.85	0.85	3161
weighted avg	0.85	0.85	0.85	3161

Afin d'améliorer les résultats, j'ai ensuite utilisé un modèle EfficientNetB1 en Transfer Learning (fine tuning sur le dernier bloc 7b). J'ai testé plusieurs hypothèses vues précédemment : recadrage, détournage ainsi que l'augmentation de données. Le meilleur résultat est obtenu sur les images non détournées avec recadrage 256x256 et augmentation de données (vertical et horizontal flip et rotation). L'accuracy est alors de 91% avec des scores en 80-85% pour les blast, ig, monocyte et smudge.

EfficientNetB1, recadrage 256x256 (entraînement 25600 images)

	precision	recall	f1-score	support
basophil	0.94	0.95	0.95	118
blast	0.82	0.89	0.86	325
eosinophil	0.99	0.97	0.98	320
erythroblast	0.96	0.91	0.94	217
ig	0.87	0.79	0.83	386
lymphocyte	0.94	0.91	0.92	467
monocyte	0.83	0.87	0.85	264
neutrophil	0.96	0.96	0.96	580
platelet	0.98	1.00	0.99	231
smudge	0.79	0.84	0.81	253
accuracy			0.91	3161
macro avg	0.91	0.91	0.91	3161
weighted avg	0.91	0.91	0.91	3161

Prédiction	basophil	blast	eosinophil	erythroblast	ig	lymphocyte	monocyte	neutrophil	platelet	smudge
Réalité										
basophil	112	0	0	1	2	0	1	0	1	1
blast	0	290	0	0	8	3	14	0	0	10
eosinophil	1	0	309	0	0	1	1	4	0	4
erythroblast	0	0	0	198	4	7	1	3	3	1
ig	6	33	0	1	306	3	17	8	0	12
lymphocyte	0	9	0	5	6	425	6	2	0	14
monocyte	0	7	0	0	10	1	230	4	0	12
neutrophil	0	2	3	0	6	3	4	559	0	3
platelet	0	0	0	1	0	0	0	0	230	0
smudge	0	11	0	0	11	11	4	4	0	212

Des tests effectués avec les modèles MobileNet et VGG16 ont donné des résultats similaires.

Le recadrage a été la solution permettant la meilleure amélioration des scores.

L'augmentation de données via transformation (flip, rotation) a eu un effet très relatif.

Le détournage n'a apporté aucune amélioration.

Difficultés rencontrées lors du projet

La principale difficulté a été d'analyser les résultats obtenus par les modèles de DeepLearning. En effet, j'ai utilisé le GradCam pour essayer de comprendre les caractéristiques identifiées par les modèles lui permettant de classer les cellules dans chaque catégorie. Cependant, cela ne m'a pas permis de les identifier.

Le temps d'apprentissage des modèles de DeepLearning (VGG16, EfficientNet....) a également été un frein. J'aurai aimé pouvoir tester plus de modèle ou faire varier plus de paramètres (taux d'apprentissage, nombre de couches et de neurones, utilisation des couches de Dropout...).

La partie traitement d'image m'a pris également beaucoup de temps, afin de réussir à trouver les algorithmes permettant de détecter le contour de la cellule.

Bilan & Suite du projet

Ce projet a été très intéressant et m'a permis de mettre en application les connaissances acquises tout au long de la formation.

Cependant, un travail en équipe aurait été plus enrichissant et peut-être plus complet.

Le résultat obtenu avec une accuracy de 91% sur la classification des 10 types de cellules est un score tout à fait correct. Cependant, étant dans le domaine médicale, un pourcentage plus élevé aurait été souhaitable.

Il serait intéressant de le rapprocher des résultats actuels obtenus par les algorithmes de traitement d'image des laboratoires pour évaluer la performance obtenue.

Pour améliorer la précision de la classification, il serait nécessaire de combiner des modèles de Deep Learning avec les algorithmes d'analyse d'images (tailles du noyau, du cytoplasme, granularité....)

Annexes

Le code source est disponible sur Github : <https://github.com/lbernard9/bloodcells/tree/main>