

# Deep Learning

Reconnaissance d'images de mangas

## Groupe de Projet

ABDOULHAMID ZAKARIA Ali

BEN AHMED DAHO Mohamed

BOUCHIKH Mouad

PAOLI Baptiste

RAVALOSON Nomanina



## Sommaire

1. Constitution de la base de données

2. Constitution du réseau de neurones

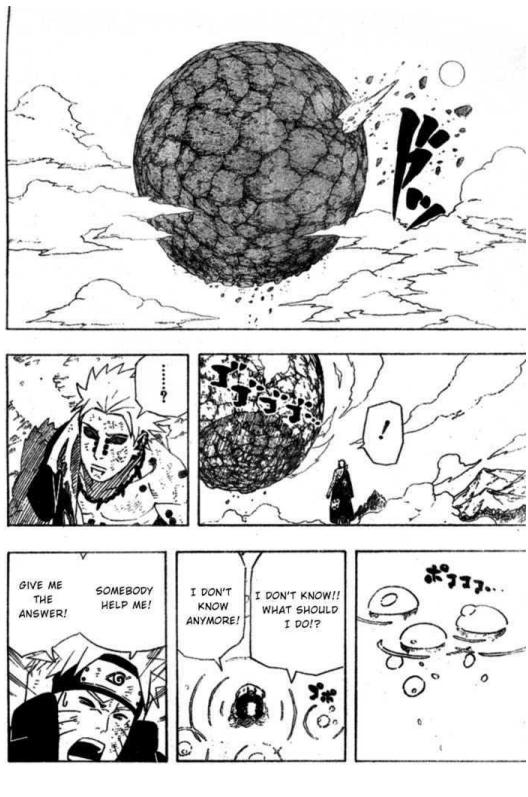
3. Analyse des résultats

4. Conclusion

# Constitution de la base de données

## Description du projet

Le but de ce projet est d'entraîner un réseau de neurones pour différencier des mangas à partir de planches. Nous avons choisis 4 mangas et avons constitué une base de 1000 images. A terme, on devrait pourvoir à partir d'une planche en entrée en déduire le manga correspondant. Les 4 mangas retenus sont : Naruto, Black Clover, Shingeki no Kyojin et Déetective Conan. Le réseau pourra s'appuyer sur des caractéristiques propres au manga comme par exemple : le style graphique, les personnages principaux ou bien le découpage des cases. Ci-dessous, voici deux exemples de planches.



## Méthodologie d'acquisition des images

Pour acquérir les images facilement, nous sommes allés sur des sites internet proposant les planches puis nous avons écrit un petit script python qui permet de renommer automatiquement les images de sorte que l'on ait qu'à glisser-déposer les images dans le dossier sans se soucier de leur nom original.

## Méthodologie de partitionnement des images

Pour chaque classe, nous disposons de 250 images. Nous avons partitionné selon la proportion suivante :

- 50% pour l'apprentissage.
- 25% pour la validation.
- 25% pour le test.

## Pronostic

On peut aisément supposer que les planches présentant les personnages principaux seront facilement reconnaissables. De plus, si l'image à reconnaître est proche d'une autre image de la base d'apprentissage, on peut supposer qu'elle sera reconnue.

## Script de chargement des données

Les données sont accessibles à : <https://github.com/mombad/DeepLearning>

Le script de chargement des données est également sur ce lien Github.

# Constitution du réseau de neurones

## Premier essai

Lors de notre premier essai, nous avions construit un réseau de neurones composé d'une couche d'entrée Dense et une couche de sortie Dense. Ce type de réseau n'étant pas adapté au problème, nous nous sommes donc intéressés à l'élaboration d'un réseau convolutif.

## Second Essai

### Principe de l'architecture

Pour notre réseau de neurones, nous avons choisi un réseau convolutif car ce type de réseau est adapté aux problèmes de traitement d'images. Le réseau est composé de couches Conv2D suivies de couches MaxPooling2D puis de couches denses.

### Fonction de coût

#### Quelle fonction de coût ?

Etant donné qu'il s'agit d'un problème de classification à 4 classes, nous avons choisi la fonction "**categorical\_crossentropy**".

#### Conséquences :

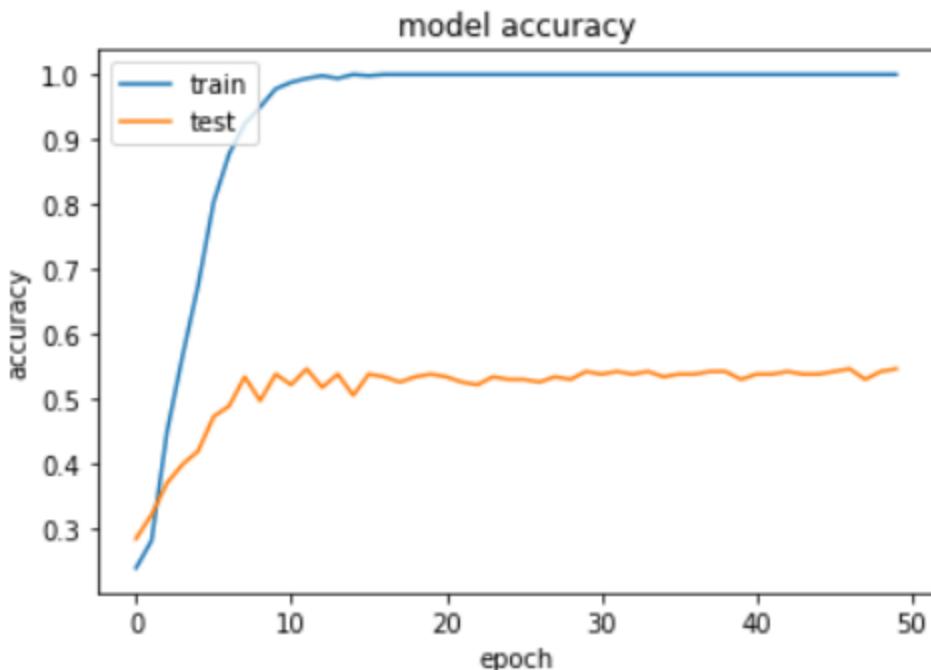
Nous avons du changer la représentation des données concernant les labels (variables **y\_train** et **y\_val**). Pour cela, il fallait passer de nombres à un tableau de taille 4 qui contient un 1 à l'index représentant la classe et des 0 ailleurs (par exemple la première classe initialement représentée par le nombre 0.0 est maintenant représentée par [1,0,0,0]).



## Problème de Surapprentissage

### Constat

Nous avons constaté un surapprentissage.



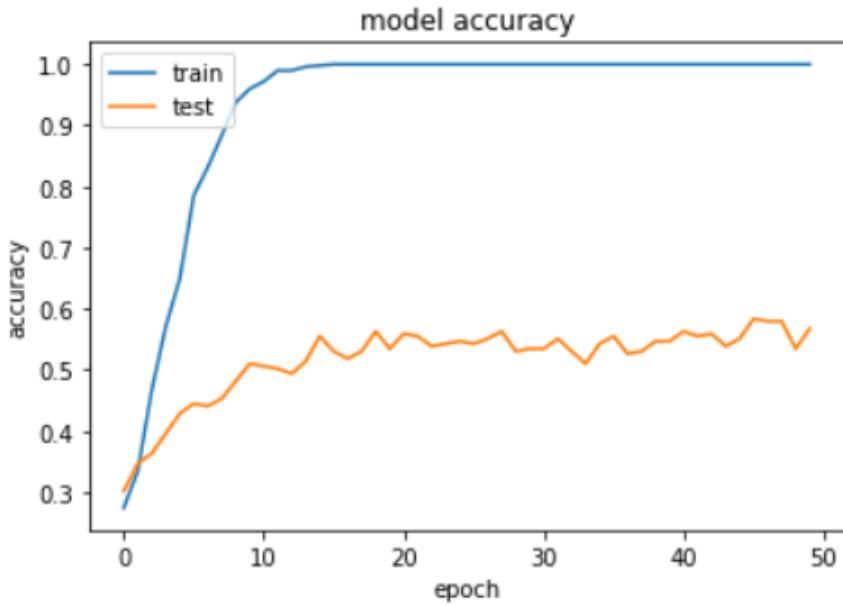
Comme on peut l'observer sur la figure ci-dessus, on voit que la courbe **train** atteint rapidement 100% alors que la courbe **test** stationne autour de 50%. Cela indique que le réseau a du mal à généraliser.

### Régularisation

Une solution simple consiste à utiliser des régularisations dans les différentes couches du réseau.



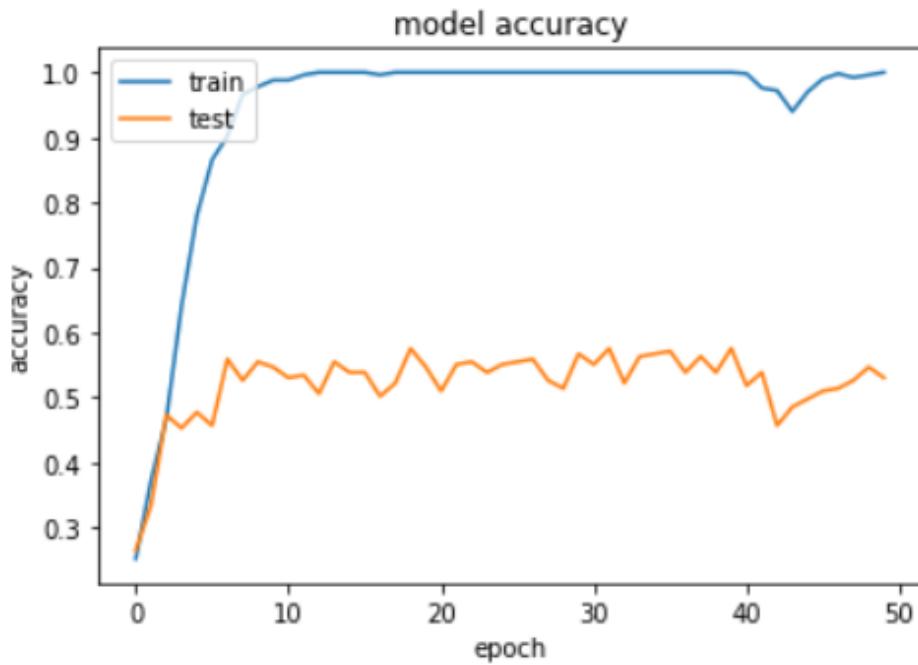
Nous avons utilisé des régularisations L2 sur les couches Conv2D et une régularisation L1 sur la couche Dense.



Le résultat est légèrement meilleur mais ce n'est toujours pas suffisant.

## Dropout

On peut utiliser une couche Dropout juste avant les couches denses pour tenter d'améliorer les résultats.

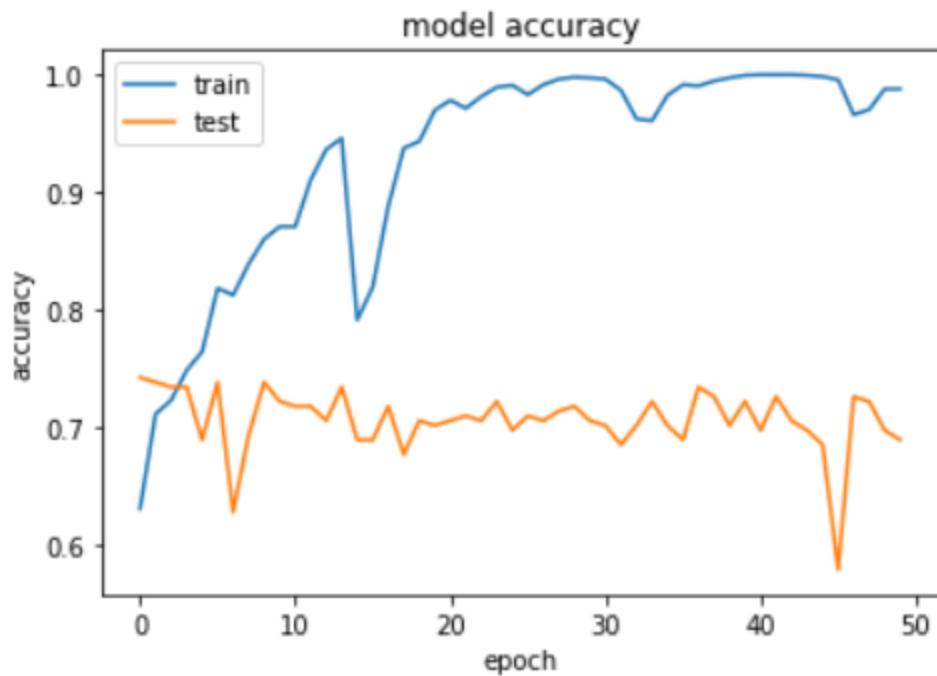




La couche Dropout ajoutée modifie l'allure de la courbe mais le résultat reste similaire.

## Augmentation de la base de données

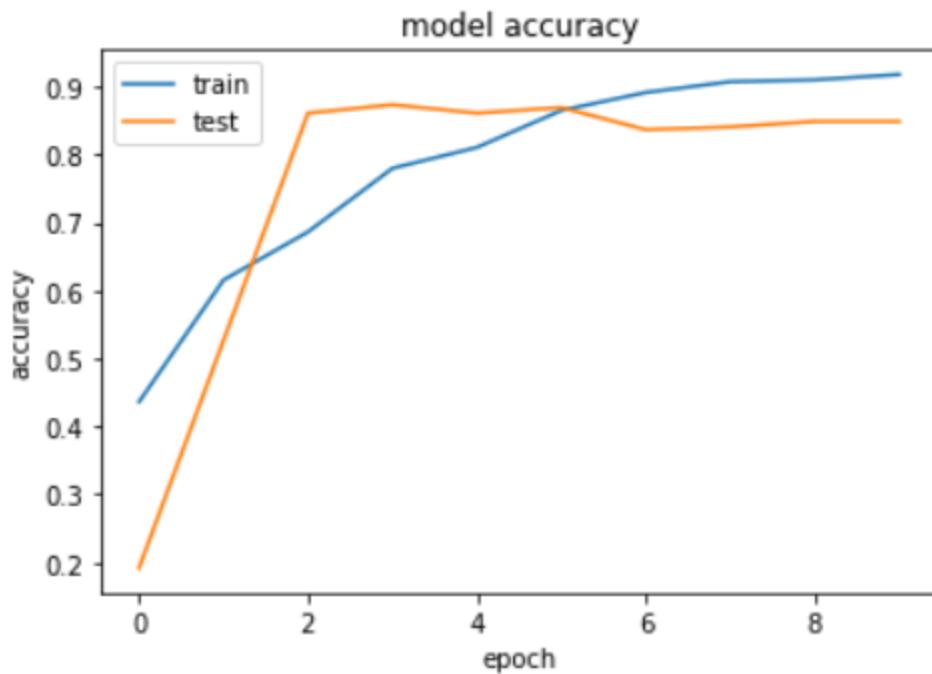
Une autre solution consiste à augmenter la taille de la base de données. Pour cela, nous avons importé des images à partir d'urls. Nous avons ajouté 1200 images à la base de données.



On obtient un résultat meilleur : environ 70%.

## Changement de réseau

Nous utilisons maintenant le réseau EfficientNet pour obtenir de meilleurs résultats. En effet, ce réseau est assez performant quand il s'agit de problèmes avec plusieurs millions de paramètres.



On obtient une accuracy à 85% ce qui est convenable.

# Analyse des résultats

## Résultats en terme d'accuracy

Nous choisissons le réseau EfficientNet comme réseau final. Comme vu précédemment, on obtient une accuracy de 85% ce qui est assez bien.

## Matrice de confusion

```
[ [51.  2.  2.  8.]
  [ 1.  61.  1.  0.]
  [ 0.  1.  62.  0.]
  [ 6.  1.  1.  56.]]
```

La matrice de confusion obtenue nous indique quels mangas ont été les plus confondus. On peut voir que Shingeki et Conan ont tous deux été peu confondus. Cela peut s'expliquer par le fait qu'ils sont assez différents des deux autres. On peut observer que le manga le plus confondu (et donc le moins bien prédict) est Black Clover avec  $2+2+8 = 12$  images mal classées.

## Accuracy par classe

On obtient ce tableau pour l'accuracy par classe.

```
[0.8095238095238095, 0.9682539682539683, 0.9841269841269841, 0.875]
```

On voit que le réseau réussit bien à identifier les planches venant de Shingeki et Conan (presque 100%) mais qu'il peine un peu pour Black Clover (80%) et Naruto (87%).

## Explications possibles

Conan est le manga le mieux prédit par le réseau. On peut expliquer cela par le fait qu'il y a beaucoup de texte sur les planches de Conan en général et que le réseau associe peut-être la présence de beaucoup de texte avec Conan. A l'inverse, Shingeki a relativement peu de texte comparé aux autres mangas et montre davantage les visages et les expressions faciales des personnages. Le style graphique est aussi assez différent.



Conan



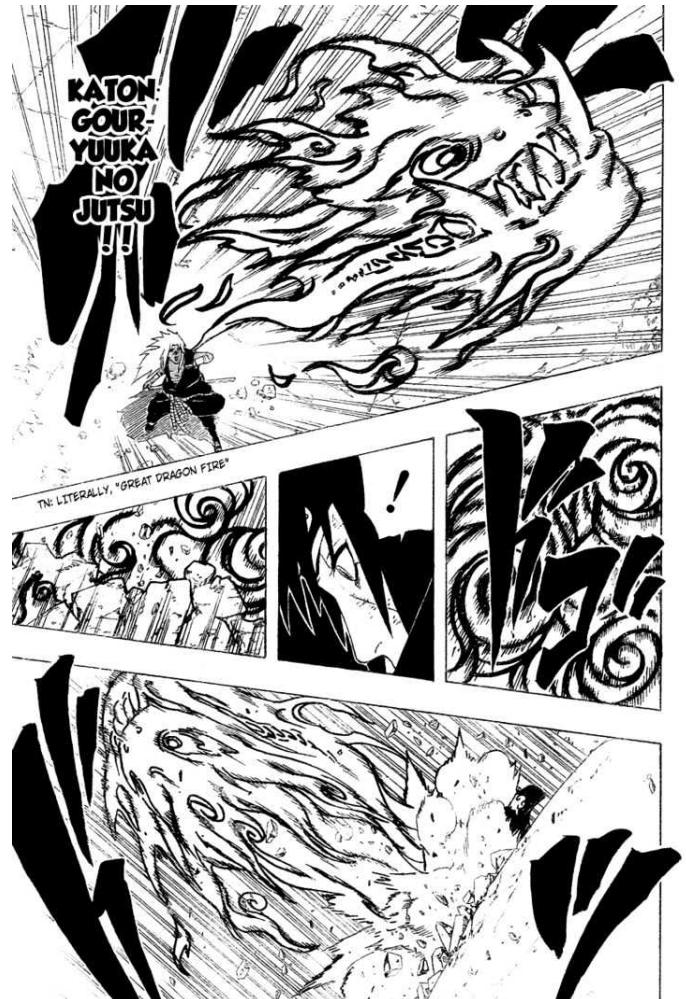
Shingeki

Ceci explique en partie pourquoi le réseau réussit plutôt bien à les différencier.

On peut remarquer sur la matrice de confusion que Black Clover et Naruto ont beaucoup été confondus entre eux. En effet, on peut voir qu'ils présentent des caractéristiques communes : un découpage de cases et un style graphique similaire ainsi que des scènes d'action. Il y a aussi beaucoup de lignes droites qui représentent la vitesse ou bien le mouvement. On peut aussi noter la présence d'onomatopées écrites en noir sur certaines images.



Black Clover



Naruto



# Conclusion

## Bilan

Ce projet nous a permis de mettre en pratique les connaissances acquises lors des cours et TPs. Nous avons pu les consolider en les appliquant à un problème de classification d'images. De plus, nous avons pu comprendre plus en détails l'intérêt de certains aspects du Deep Learning. Nous avons aussi pu expérimenter le réseau EfficientNet.

Nous avons obtenu un résultat final satisfaisant mais qui peut encore être amélioré. On peut augmenter la taille de la base de données. On pourrait augmenter la taille des images pour avoir une meilleure qualité. Pour compenser le temps de traitement supplémentaire on peut importer les images en nuances de gris au lieu de RGB. Cela diminuerait par 3 la taille d'une image.