# Projet S8

## Reconnaissance de styles de peintures

June 8, 2023

---

Cyprien Fourcroy
Etienne Andrier
Eliott Py
2ème annéee cursus ingénieur

Enseignants : Jean-Luc Collette, Michel Ianotto

---

CentraleSupélec

## 0.1 Base de Donnée

Pour notre base de donnée nous somme partis du travail de WikiArt, une encyclopédie d'art qui a en particulier créer un jeux de données de peintures réparties sur 14 styles différents. Cette base de données est une version plus "légère" d'une autre base de donnée de référence, qui elle contient 28 styles ainsi que des informations sur l'auteur et la période de la peinture. Nous avons donc choisit cette database pour son nombre important de peintures, et un nombre de classes qui nous permettait d'explorer plusieurs modèles différents.

| | |
|---|---|
| 1 | abstract |
| 2 | animal-painting |
| 3 | cityscape |
| 4 | figurative |
| 5 | flower-painting |
| 6 | genre-painting |
| 7 | landscape |
| 8 | marina |
| 9 | mythological-painting |
| 10 | nude-painting-nu |
| 11 | portrait |
| 12 | religious-painting |
| 13 | still-life |
| 14 | symbolic-painting |

Figure 1: Les 14 styles de la base de donnée

Cependant, cette base de donnée présentait quelques défauts comme la présence de peintures ambigus (dont même les membres du projet ne pouvait pas donner le genre). Le problème principal reste la répartition inégale des peintures dans les 14 classes, avec un facteur 10 entre les plus présentes et les moins peuplées. Nous avons donc choisi de nous consacrer sur les 6 classes avec le plus d'éléments : peintures abstraites, de genre, religieuses, les portraits ainsi que les paysages urbains et ruraux.

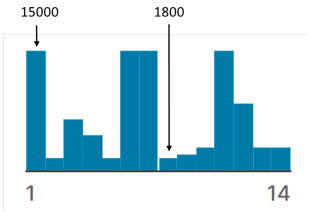


Figure 2: Nombre de peintures pour chaque classe

## 0.2 État de l'art

La majorité des modèles récents de classification d'images fonctionnent à partir de réseaux de convolutions, et ArtGraph en fait partie. C'est actuellement l'architecture avec les meilleures performances sur la détection de genre des peintures. Il fonctionne à partir de 2 sous-réseaux : le premier, ResNet50, extrais des informations de la peinture. Le second réseaux utilise un arbre de relations entre les artistes (leurs mouvements, professeurs et leurs influences) et les peintures (Genre de la peinture, période, ville de vente/réalisation). De cet arbre le réseaux node2vec extrait du contexte pour notre peinture. Les 2 sources d'information sont ensuite concaténées dans un encodeur. C'est bien cet encodeur qui est entraîné dans cet article.

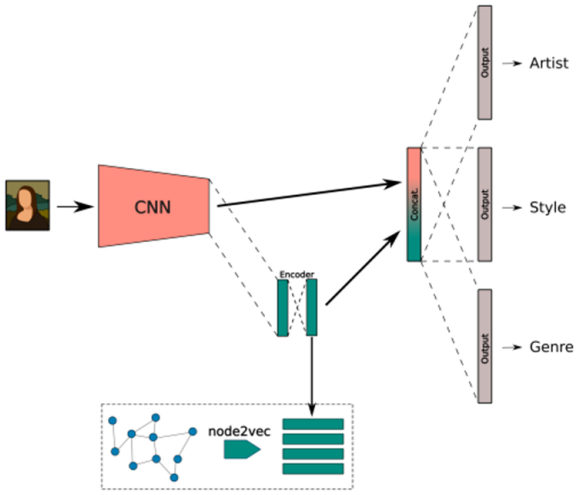


Figure 3: Architecture de ArtGraph

En s'entraînant sur 28 genres de peinture, ArtGraph arrive à atteindre 66.52% de classifications correctes, contre 65% pour un ResNet qui aurait été fined-tuned sur des genres de peintures.

## 0.3 Réseaux convolutifs

La méthode des réseaux convolutifs a été rendue populaire en 1998 par Yann le Cun avec son réseau LeNET-5, composé de deux couches de convolution, deux couches de pooling et d'une couche entièrement connectée. Ce réseau avait pour vocation de réaliser la tâche de la reconnaissance de chiffres, et a donc été entraîné sur la base de données MNIST, dont les éléments sont de dimension (28,28). Nous avons voulu entraîner un modèle de réseau convolutif sur la base Wikiart. Pour cela nous avons dans un premier temps évalué la faisabilité de cette tâche, nous avons donc choisi deux classes qui sont très

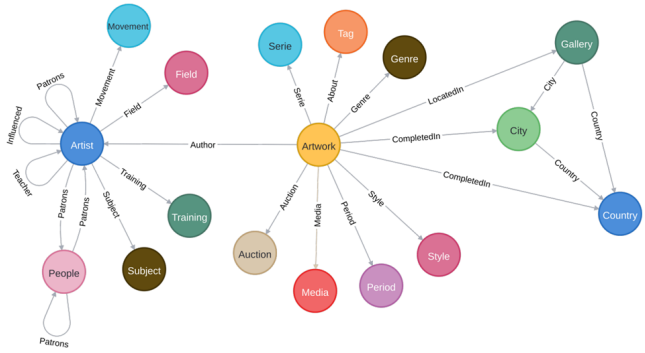


Figure 4: Arbre de connaissance, où les noeuds correspondent à des peintures et les arêtes à des relations existantes

facilement distinguables : les paysages et les portraits. Dans un premier temps, nous avons utilisé des images de taille (32,32), car nous avions une contrainte de puissance de calcul. La première architecture que nous avons utilisée est un réseau de cinq couches de convolution et une couche de pooling, chaque couche de convolution étant suivie d'une couche de *batch normalization* Notre manque d'expérience à ce stage du projet explique cette architecture étrange, qui diffère du schéma classique où une couche de convolution est toujours suivie d'une couche de pooling.

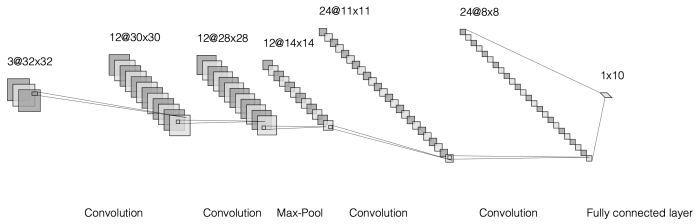


Figure 5: Premier modèle convolutif

Nous avons, avec 13 000 données d'entraînement et 2000 données de test par classe, obtenu 90% de précision sur les deux classes paysage et portrait. Nous avons par la suite étendu le modèle aux 6 classes qui contiennent le plus d'éléments, soit *abstrait*, *paysage*, *paysage urbain*, *peinture de genre*, *portrait*, *peinture religieuse*. Nous avons attribué 4000 images à l'entraînement et 2000 au test pour chaque classe. Nous avons obtenu, sans changer de structure de réseau et avec une taille d'image de (32,32), une précision de 50%. Après avoir réussi à maîtriser le calcul sur GPU nous nous sommes permis d'augmenter la taille

des données et le nombre de classes. Par ailleurs nous avions à ce moment une meilleure compréhension des réseaux convolutifs, et nous avons donc pu en créer un plus adapté à nos données. Nous avons utilisé une structure qui se rapproche de celle de LeNET-5, mais avec plus de couches et avec la fonction d'activation ReLu (au lieu d'un sigmoïde). Notre réseau est donc composé de trois couches de convolution, trois couches de pooling et trois couches entièrement connectées. Nous avons également testé la normalisation en sortie de chaque couche, mais se changement de structure s'est accompagné d'une chute de la performance. Nous avons expérimenté avec différentes configurations : des tailles de filtre croissantes ou décroissantes à mesure que l'on s'avance dans le réseau, des petites tailles de filtre (3 par 3) ou des grandes tailles (11 par 11), des nombres de filtres constants ou croissants. Nous avons entraîné ce modèle sur 14 classes, et un

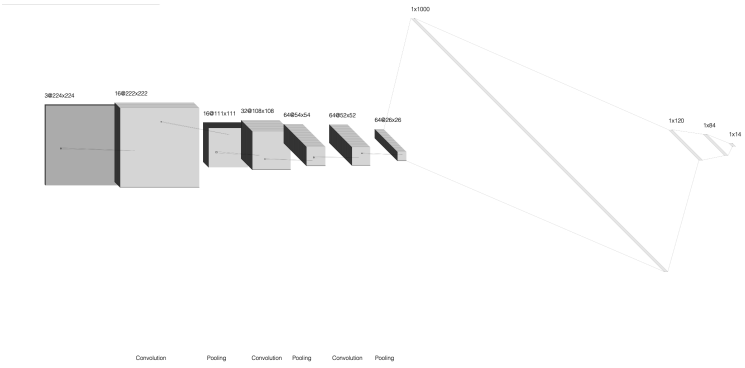


Figure 6: Deuxième modèle convolutif

total de 85 000 images, pour environ 15 000 images de test, ces images sont de tailles (224,224). Nous sommes parvenus à maintenir la performance du premier modèle sur 6 classes, soit 50% sur 14 classes.

## 0.4 Resnet18

Les structures Resnet ont été développées à partir de 2015 afin d'améliorer les performances des classificateurs d'images. La principale idée est d'augmenter la profondeur des réseaux de convolution en introduisant des skip connections afin de régler le problème du vanishing gradient. En effet, lors de l'actualisation des poids qu'on effectue pendant la rétropropogation, le gradient dimininue à mesure que l'on progresse vers les couches initiales. Il existe de nombreux modèles de Resnet avec des modèles plus ou moins profonds. Nous avons utilisé le modèle Resnet18 pour notre problème, 18 étant le nombre de couches. Les modèles les plus connus sont déjà fournis dans la bibliothèque Pytorch. En plus d'une architecture déjà prédéfinie, le modèle est aussi préentraîné sur une base de donnée avec plus ou moins de classes, 1000 classes dans notre cas. Lorsque le modèle est préentraîné, il est en mesure de détecter des caractéristiques particulières aux images. Les images sur lesquelles il est préentraîné sont celles d'animaux, de bâtiment et d'objets divers, images dont les caractérisques peuvent être retrouvées dans notre base de donnée. Ainsi, on peut espérer que Resnet18 préentraîné est en mesure de détecter les caractéristiques de nos peintures. Il faut cependant que nous adaptions le réseau à notre problème ce qui peut être réalisé en modifiant la dernière couche. Elle doit renvoyer un vecteur ayant comme nombre de compoosants celui de classes (14. On doit aussi l'entraîner sur notre base de données.

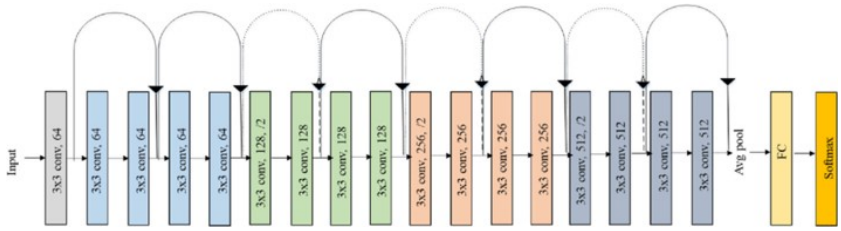


Figure 7: Structure de Resnet18

L'entraînement du réseau reste long car l'image de 224x224 subit un traitement complexe avant d'être envoyée à la dernière couche. L'actualisation des poids de la dernière couche et la calcul du gradient de la fonction de perte ne sont pas les étapes les plus longues. Quelques epochs suffisent pour que le modèle converge en utilisant un optimizer SGD.

## 0.5 Conclusion

Nous sommes heureux d'avoir pu travailler sur ce problème exigeant, de par la subtilité des différences entre les classes que nous avons à distinguer et de par la richesse d'information que possèdent les images. Nous ne sommes pas parvenu à atteindre les mêmes performances que l'état de l'art, et la performance des modèles à base de réseaux convolutifs a plafonné à 15 points

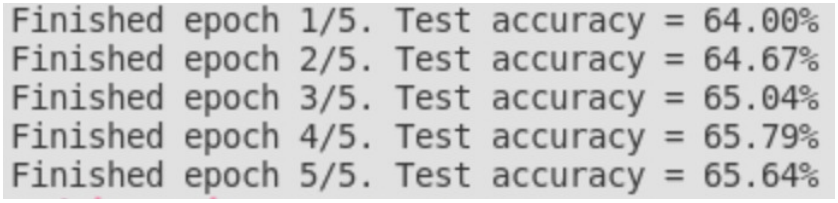


Figure 8: Résultat du Resnet18

du modèle basé sur ResNET. Différentes pistes d'amélioration existent pour améliorer ces chiffres. Une première piste d'amélioration est d'implémenter un algorithme d'optimisation des hyperparamètres, comme la quantité d'images d'entraînement, le nombre et la taille des filtres etc. Nous aurions pu également proposer, lorsque l'ambiguïté existe, deux classes les plus probables pour une image, et ainsi réévaluer la performance de notre modèle. Enfin, il aurait été possible mais coûteux en temps (la base de données contient 80 000 images) de nettoyer la base de données, car nous avons remarqué certaines erreurs. Enfin, et la faisabilité de cette méthode reste questionnable, nous aurions pu entraîner les couches cachées de ResNET en plus de la couche entièrement connectée.

Nous remercions Michel Ianotto et Jean-Luc Collette pour leurs temps ainsi que les conseils qu'ils nous ont donnés.