

# Comparaison de réseaux pour la reconnaissance de style de tableaux

CentraleSupélec ISIA 2018 - Projet Deep Learning

[https://github.com/ChloeGobe/art\\_deep\\_ecp](https://github.com/ChloeGobe/art_deep_ecp)

Chloé Gobe

chloe.gobe@student.ecp.fr

Xavier Rettel

xavier.rettel@student.ecp.fr

Mounia Slassi

mounia.slassi-sennou@student.ecp.fr

## Abstract

Prendre connaissance du style d'un tableau est un premier pas important vers la compréhension d'une oeuvre. Son style permet de la situer dans un contexte historique et de mieux saisir les caractéristiques et composants du tableau. Néanmoins, la détermination de certains styles de peinture, au-delà des œuvres provenant d'artistes qui se sont constitués en écoles, relève parfois d'une labelisation subjective bien que fondée sur de puissantes critères déterminés par les spécialistes. Dans ce document, nous nous sommes intéressés à reproduire les travaux de chercheurs sur les déterminations des styles de tableaux à l'aide de Convolutional Neural Networks (CNNs) en essayant différents réseaux afin de vérifier si les résultats obtenus permettaient de retrouver des proximités artistiques et historiques entre différents mouvements. Pour cela, nous avons tout d'abord essayé de faire du transfert learning et du fine tuning avec différents réseaux, puis avons choisi le plus performant (ResNet50) et avons essayé d'améliorer nos performances avec des techniques de régularisation. Nous aboutissons à des résultats encourageants, à savoir une précision de 50.46% sur 25 styles. Enfin, nous avons essayé de vérifier si les parentés historiques entre les différents styles de peinture étaient repérées par le réseau entrainé.



FIGURE 1: Exemples de tableaux de styles différents (de gauche à droite) : *Le Voyageur contemplant une mer de nuages* de Friedrich, *La nuit étoilée* de Van Gogh et *La laitière* de Vermeer.

## 1. Introduction

La numérisation des œuvres d'art a permis à un public large d'apprécier des œuvres de tout type sans avoir besoin de se rendre dans des musées, parfois à l'autre bout du monde. L'étude et la compréhension des chefs d'œuvre de l'humanité ne se sont trouvés que facilités par les bibliothèques grandissantes d'œuvres numérisées. Des institutions comme le Metropolitan Museum de New York ou encore le Rijksmuseum d'Amsterdam ont numérisés de nombreuses œuvres de leurs collections.

L'être humain étant prolifique lorsqu'il s'agit d'exercer sa créativité et son imagination, le nombre d'œuvres disponibles est devenu important, donnant ainsi la possibilité aux chercheurs de travailler sur ces nouvelles bases de données avec pour buts, une meilleure compréhension de l'art et de notre Histoire et une capacité à mieux connaître et comprendre la peinture. Il est maintenant possible de détecter les contrefaçons grâce à de l'apprentissage supervisé ou reconstruire le style de certains artistes sur les photos prises aujourd'hui.

L'étude de tableaux est un art en soi, tant abondent les œuvres dans le temps et dans l'espace. Il est nécessaire de bien connaître les différentes matières, caractéristiques, courants, artistes mais aussi l'Histoire de l'Art. Partant de ce constat, il est difficile, pour un non-initié, de correctement appréhender une œuvre et de la situer dans un courant artistique. Le travail de *Lecoutre et al*[6] a néanmoins permis, grâce au deep learning, de créer une application web[7], étant capable de donner le style d'une peinture ou les styles s'en approchant le plus.

Toutefois, les styles artistiques ne sont parfois que des labels, l'artiste ne se base pas sur les caractéristiques de chaque style pour créer ses œuvres. Ce sont les spécialistes ou les contemporains, qui à posteriori vont essayer de déf-

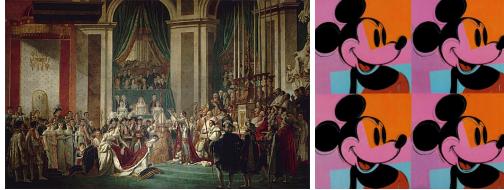


FIGURE 2: Exemple de tableaux de styles différents : à gauche *Le sacre de Napoléon* de David, à droite *The Art of Mickey Mouse* de Warhol.

nir des courants artistiques, des styles de peinture, à partir des œuvres peintes, en essayant de définir des caractéristiques clés de chaque label trouvé.

Nous nous sommes donc demandé si le deep learning nous permettait de retrouver les catégories de style posées par les experts à partir de bases de données de tableaux numérisés. Dans un deuxième temps, nous aimerions savoir si, au-delà des proximités temporelles et spatiales connues, l'aspect des tableaux eux-mêmes nous permettrait de retracer ces liens entre les différents courants artistiques.



FIGURE 3: Exemple de tableaux de style différents : à gauche *Flower thrower* de Banksy, à droite *La Naissance de Vénus* de Botticelli

## 2. Etat de l'art

Nous avons pu trouver plusieurs études qui s'intéressent au problème de la reconnaissance de style des tableaux, chacune abordant la problématique de manière différente.

La première étude que nous avons pu analyser est "*Ceci n'est pas une pipe : A Deep Convolutional Network for Fine-art Painting Classification*"[9], qui a été réalisée par des étudiants de l'Université de Shinshu (Nagano, Japon), et de l'Université de Malaya (Kuala Lumpur, Malaisie). La deuxième étude que nous avons examinée est "*Recognizing Art Style Automatically in painting with deep learning*"[6], rédigée par des chercheurs de l'INSA de Rouen et de l'Université Paris Dauphine.

Les deux études ont sensiblement le même objectif : reconnaître le style d'un tableau à partir de la base de données Wikipaintings.

"*Ceci n'est pas une pipe : A Deep Convolutional Network for Fine-art Painting Classification*"[9] utilise comme

unique dataset Wikipaintings qui est découpé en trois sous-éléments : training, validation et test. L'architecture du CNN utilisée est fortement inspirée d'AlexNet : le réseau possède 5 couches de convolution, 3 couches max-pooling et 3 couches complètement connectées. Le réseau prend en entrée des images de taille 227 x 227 x 3. Le réseau a été pré-entraîné sur ImageNet et la dernière couche a ensuite été ré-entraînée. Il a été montré que la précision entre des classificateurs Softmax et SVM sont similaires. Le résultat de l'étude montre qu'utiliser du fine-tuning sur réseau CNN pré-entraîné sur ImageNet est bien plus performant qu'un réseau entièrement entraîné sur le dataset.

*"Recognizing Art Style Automatically in painting with deep learning"*[6] utilise deux datasets : Wikipaintings pour les parties training, validation et test et en supplément, la base de données ErgSap pour la phase de test, afin d'avoir une base de données indépendante, qui a été labellisée par des experts indépendants de la communauté WikiArt. Les réseaux utilisés sont AlexNet et Resnet50. Les réseaux sont pré-entraînés sur ImageNet et les recherches des scientifiques ont démontré que les résultats étaient meilleurs si 20% des couches étaient ré-entraînées au lieu de seulement la dernière. Du bagging, du dropout et de la data augmentation sont utilisés.

## 3. Dataset

Pour réaliser notre projet, nous nous sommes servis de la même base de données que l'équipe de RASTA[6]. Il s'agit d'un dataset issu de Wikiart[2] (anciennement Wikipaintings), qui contient 82 154 images de tableaux, réparties selon 25 styles différents de peinture (entre parenthèses le nombre d'images total dans chaque catégorie) :

- Abstract Art (1006)
- Abstract Expressionism (2091)
- Art Informel (969)
- Art Nouveau (4163)
- Baroque (3992)
- Color field Painting (1258)
- Cubism (1686)
- Early renaissance (1330)
- Expressionism (5844)
- High Renaissance (1275)
- Impressionism (12164)
- Magic Realism (1011)
- Mannerism (1192)
- Minimalism (1267)
- Naive Art (2026)
- Neoclassicism (2700)
- Northern Renaissance (2405)
- Pop Art (1129)
- Post-Impressionism (5998)
- Realism (10090)

- Rococo (1928)
- Romantism (6569)
- Surrealism (4799)
- Symbolism (4063)
- Unikiyo-e (1178)

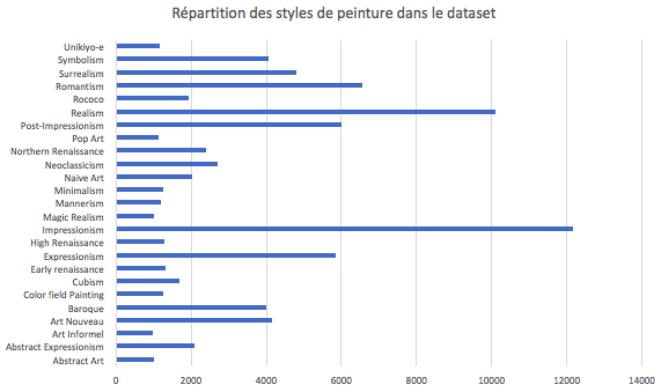


FIGURE 4: Répartition des styles dans le dataset

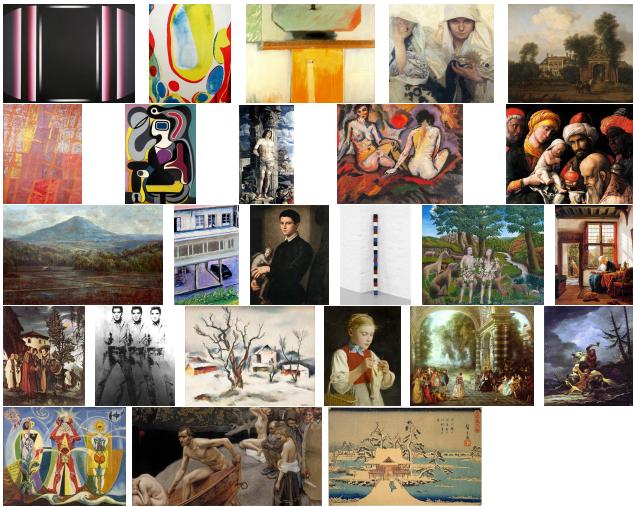


FIGURE 5: Exemples d'images du dataset Wikipaintings

Le dataset a été constitué en sélectionnant, parmi les nombreux styles disponibles sur Wikiart, ceux pour lesquels il y avait un nombre relativement conséquent d'oeuvres disponibles. Chaque catégorie dispose ainsi d'au moins environ 1000 images.

Le dataset est découpé en trois sous-parties, qui correspondent respectivement aux ensembles d'entraînement, de validation et de test :

- *wikipaintings-train* qui contient 66 549 images
- *wikipaintings-validation* qui contient 7 383 images
- *wikipaintings-test* qui contient 8 222 images

Nous avons ainsi environ 81% des tableaux qui servent à l'entraînement, 9% pour la validation et 10% pour le test. La répartition des différentes catégories au sein de chaque sous-ensemble est la même que celle du dataset complet.

Nous réutilisons ce dataset, d'une part par souci de simplicité, d'autre part dans une optique de comparaison avec les résultats trouvés par l'équipe de RASTA.

Nous attirons l'attention du lecteur qui souhaiterait reproduire nos expériences sur la taille relativement conséquente du dataset, à savoir près de 20Go.

## 4. Approche

Nous souhaitons ici partir d'une image d'un tableau et arriver à la classifier selon son style. Cette tâche étant tournée vers la vision par ordinateur, et au vu des approches mises en place dans les articles mentionnés précédemment, nous nous sommes orientés vers l'utilisation de réseaux neuronaux convolutifs (CNN).

Nous avons essayé plusieurs réseaux présents dans Keras, qui se trouvent dans le package *keras.applications*.

Nous avons décidé d'utiliser un protocole similaire à [6] et [9] concernant l'entraînement de nos réseaux, et nous nous sommes ainsi basés sur les réseaux de Keras pré-entraînés sur ImageNet.

En effet, [9] a démontré que de meilleurs résultats étaient alors obtenus, et ce de manière beaucoup plus rapide, en comparaison avec un entraînement complet d'un réseau sur le dataset de tableaux.

Nous avons alors adopté une démarche de *transfert learning* et de *fine tuning*. Les réseaux que nous avons utilisé (listés plus loin) sont en effet plutôt adaptés à la reconnaissance d'objets.

Nous avons sélectionné quelques réseaux *a priori* d'intérêt, notamment de par leur utilisation dans l'état de l'art, mais aussi de par leur taille relativement contenue, qui correspondait bien aux ressources assez faibles dont nous disposions.

Afin d'adapter leur utilisation à notre cas d'usage, nous leur avons enlevé leur dernière couche, qui correspond à la sortie qui indique quel objet a été reconnu, et l'avons remplacée par une autre couche de sortie de type *softmax* à 25 neurones (un pour chaque style) précédée par une couche d'*average pooling*.

Nous avons tout d'abord pensé n'entraîner que la dernière couche, mais vu les performances limitées obtenues, nous avons décidé d'autoriser la mise à jour des poids d'autres couches.

Nous avons alors utilisé les résultats fournis par [6] qui indiquent en effet que l'entraînement des dernières couches du réseau permettent d'augmenter significativement les performances de classification. Le chiffre de 20% des couches à mettre à jour semble émerger de leurs travaux, et nous

avons ainsi décidé de rendre les derniers 20% des couches de nos réseaux "entraînables" dans le graphe Keras.

L’entraînement a été réalisé en utilisant une taille de batch de 64, l’optimiseur Adam et la fonction de coût *categorical crossentropy*.

Une fois l'entraînement réalisé sur les réseaux sélectionnés, nous nous sommes concentrés sur celui qui a présenté les meilleures performances et nous avons tenté d'améliorer ces dernières.

Nous avons alors mis en place des techniques de régularisation. Nous avons en effet essayé de faire de la *data augmentation* grâce à fonctionnalités de pre-traitement de Keras.

Notre réflexion a porté sur ce qui pouvait renfermer le style d'un tableau et sur quelles transformations pouvaient le laisser invariant.

De manière intuitive, nous nous sommes dit que le retournement horizontal conservait le style.

Un retournement vertical peut cependant éventuellement altérer le style, de même qu'une rotation.

Nous avons enfin considéré que le zoom ne modifiait pas le style d'un tableau : un tableau observé de près ou de loin est toujours le même tableau.

Toujours dans une optique de régularisation, nos entraînements se sont déroulés en utilisant une fonctionnalité de Keras qui implémente l'*early stopping*.

Nous avons enfin construit des matrices de confusion afin de mieux comprendre les résultats fournis par les différents réseaux et afin d'obtenir des comparaisons avec les articles mentionnés. Nous avons en effet cherché à savoir si les styles de peinture qui se distinguaient bien lors des expériences des chercheurs se distinguaient de même avec nos essais.

## 5. Expériences

### **5.1. Protocole**

N'ayant pas de GPU sur nos machines et nos calculs nécessitant a priori beaucoup de puissance, nous avons utilisé une instance à la demande Amazon Elastic Compute Cloud (EC2) particulièrement adaptée au calcul GPU dont voici les caractéristiques :

- Type de l’instance : p2.xlarge
  - GPU : Nvidia K80 (12Go)
  - Version de Keras : 2.1.4
  - Version de Tensorflow : 1.6

Tous les calculs ont tourné au sein d'un environnement virtuel Conda fourni et pré-configuré par Amazon[8].

Nous avons décidé de comparer les performances de plusieurs réseaux de neurones, en nous intéressant tout particulièrement à :

- Inception V3
  - Xception
  - InceptionResNet V2
  - NasNet Mobile
  - ResNet50

Sur ces 5 réseaux de neurones, notre approche a été sensiblement la même. Nous avons tout d'abord récupéré les poids correspondants à chacun de ces réseaux pré-entraînés sur le dataset ImageNet grâce à Keras.

Etant donné les spécificités des tableaux, nous avons ensuite décidé de réajuster les poids en réentraînant, dans deux scénarios différents, soit la dernière couche, soit les dernières couches (correspondant à 20% des couches), sur le data set WikiPainting, que nous avons récupéré via le repository Github des chercheurs de "*Recognizing Art Style Automatically in painting with deep learning*".

**Inception V3** Le réseau Inception V3[5] a une profondeur de 159 et possède 23,851,784 paramètres. L'architecture de Inception V3 peut être résumé à :

- des entrées de taille 299x299x3
  - des couches de convolution et de max-pooling pour convertir les images en des représentations 8x8x2048
  - un average pooling pour se ramener à une taille de 1x1x2048

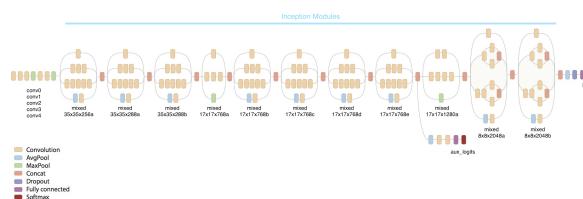


FIGURE 6: Schéma du réseau InceptionV3

**Xception** Le réseau Xception[3] a une profondeur de 159 et possède 23,851,784 paramètres. Il correspond à un réseau Inception dont les modules d’Inception ont été remplacés par des convolutions séparables en profondeur. Ce réseau est *a priori* légèrement meilleur que Inception sur ImageNet.

**InceptionResNet V2** Le réseau InceptionResNet V2 a une profondeur de 572 et possède 55,873,736 paramètres.[1]

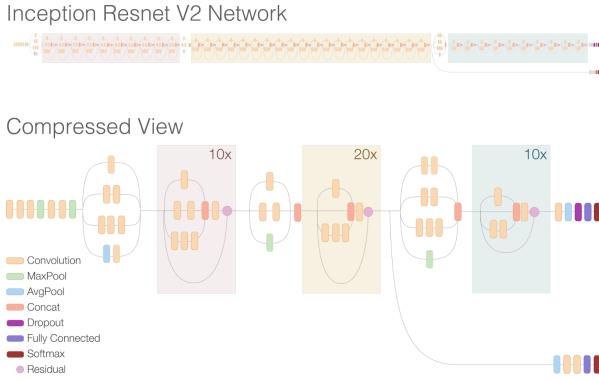


FIGURE 7: Schéma du réseau InceptionResNet V2

**NasNet Mobile** Le réseau NasNet Mobile[10] (Neural Architecture Search Network) est un réseau proposé récemment par les équipes de Google. Il s'agit de la version allégée du réseau NasNet, qui est notamment connu pour avoir été généré à l'aide d'outils de machine learning.

**ResNet50** Le réseau ResNet50[4] a une profondeur de 168 et possède 25,636,712 paramètres. Il s'agit d'un réseau résiduel à 50 couches.

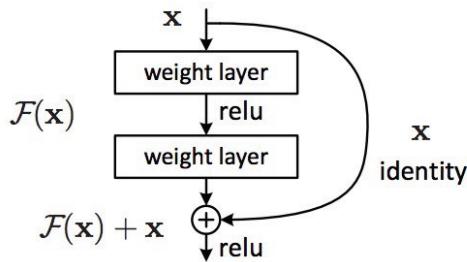


FIGURE 8: Block ResNet

Nous avons réentrainé les 33 dernières couches et mis un GlobalAveragePooling et une couche de sortie softmax

## 5.2. Résultats

**Essais de différents réseaux** Présentés ci-dessous, les précisions Top-1, Top-3 et Top-5 obtenues sur l'ensemble de test après entraînement suivant le réseau de base employé. Tous les résultats correspondent aux réseaux où les 20% des dernières couches ont été entraînés, sauf la dernière ligne, qui présente les résultats pour ResNet50 avec uniquement la dernière couche entraînée.

Malgré plusieurs tentatives de modification des réseaux retournant des résultats plus faibles, nous n'avons pas réussi à obtenir des résultats aussi intéressants que ceux

	Top-1	Top-3	Top-5
InceptionV3	12.30%	14.96%	18.68%
Inception ResNet V2	14.49%	17.46%	21.18%
NasNet Mobile	11.34%	26.17%	39.25%
Xception	12.30%	15.05%	20.61%
ResNet50	50.46%	75.89%	86.53%
ResNet50 (dernière couche)	34.65%	61.38%	72.10%

FIGURE 9: Résultats obtenus sur le dataset WikiPaintings avec différents réseaux disponibles sur Keras

obtenus avec ResNet50. Nous constatons également bien une amélioration forte des performances du réseau en entraînant plus de couches.

En traçant la matrice de confusion d'un des réseaux, nous ne sommes rendus compte que le réseau donnait toujours le même style de peinture en sortie du réseau, peu importe l'entrée.

Nous nous sommes interrogés sur ce phénomène qui semble toucher le reste de nos réseaux. Les hypothèses que nous avons pu avancer est la présence d'un minimum local dans la fonction de coût, ou un problème d'hyperparamètres que nous n'avons pas pu résoudre. Nous n'avons malheureusement pas pu explorer plus en profondeur ce souci.

Au vu des résultats, pour la suite de l'étude, nous avons donc choisi d'améliorer les résultats obtenus avec ResNet50 en modifiant plusieurs paramètres.

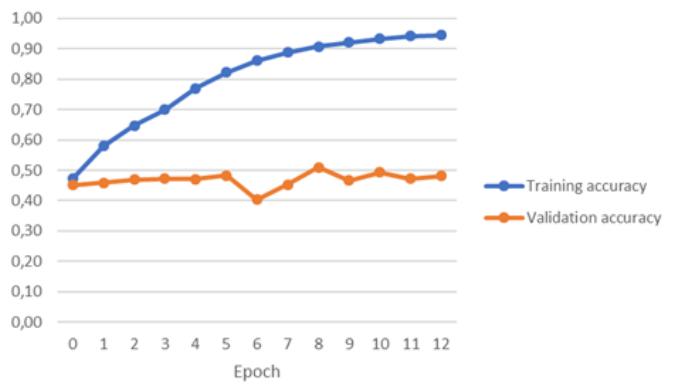


FIGURE 10: Courbes obtenues grâce à Tensorboard lors de l'entraînement à partir du réseau ResNet50.

**Amélioration de ResNet50** Voici les résultats obtenus sur la base de ResNet50 avec les modifications suivantes :

- Data Augmentation
- Zoom : entre 80% et 120%

— Rotation : entre -90 et +90 degrés

	<b>Top-1</b>	<b>Top-3</b>	<b>Top-5</b>
De base	50.46%	75.89%	86.53%
Ajout zoom	51.19%	77.5%	88.01%
Ajout zoom et rotation	50.53%	77.60%	87.68%

FIGURE 11: Résultats de précision sur l’ensemble de test obtenus avec ResNet50 après amélioration

On constate une amélioration de quelques pourcents en utilisant la data augmentation.

On remarque cependant que la rotation a plutôt l’air de diminuer les performances. Comme conjecturé dans la partie approche, il semble que le style d’un tableau ne soit pas forcément complètement insensible à son orientation.

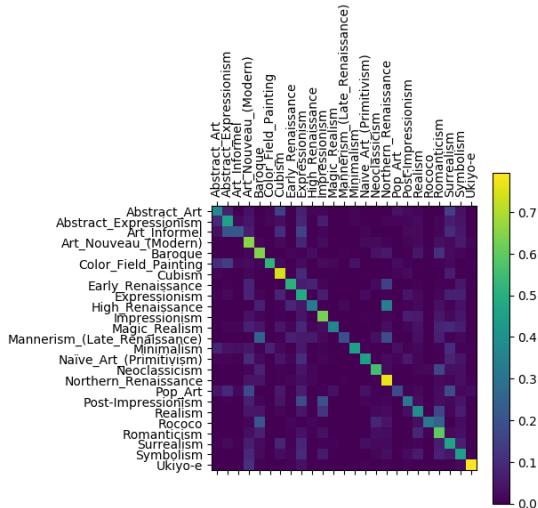


FIGURE 12: Matrice de confusion obtenue avec ResNet50

**Interprétation** Nous pouvons constater que les styles de peinture sont plutôt bien reconnus par le réseau basé sur ResNet50. Les styles Ukiyo-e et Cubisme sont reconnus à plus de 70% car en effet, ces styles sont très particuliers et distincts des autres.



FIGURE 13: Cubisme



FIGURE 14: Ukiyo-e

Le modèle a des difficultés à distinguer des styles de peinture très similaires, ce qui permet notamment de retrouver une chronologie dans les différents courants de peinture historique. Nous pouvons ainsi rapprocher le Baroque et le Mannerism, deux courants proches du XVIème siècle, ou encore les trois styles de Renaissance.



(a) Northern Renaissance



(b) Late Renaissance



(c) High Renaissance

FIGURE 15: Exemples de tableaux des différents courants de la Renaissance

Une critique peut potentiellement également être émise sur notre dataset. Il est en effet issu de Wikiart, où les tableaux sont labellisés à la main par des internautes potentiellement amateurs. Il peut ainsi y avoir quelques erreurs de labellisation.

## 6. Conclusion

Nous avons vu qu’il était possible, grâce au deep learning, de reconnaître avec une certaine précision le style de tableaux artistiques.

Le réseau ResNet50 s’est montré particulièrement efficace pour cette tâche en adaptant sa dernière couche pour notre problème et en ré-entraînant ses dernières couches.

Nous avons également vu que la data augmentation avait une influence bénéfique sur nos performances, même s’il n’est pas forcément évident *a priori* que les transformations possibles conservent le style d’un tableau.

Nous voulons également appuyer sur l’intérêt de ce problème de classification. Il est en effet essentiel de pouvoir catégoriser un tableau, notamment dans le cadre d’un recensement dans un musée. Si ce processus est aujourd’hui réalisé fastidieusement à la main par des personnes hautement qualifiées, il pourrait demain être simplifié par des approches type *deep learning*.

Enfin, nous avons vu que notre réseau a parfois du mal à distinguer certains styles. On peut argumenter que le style d'un tableau est relativement subjectif. Certains se sont superposés géographiquement ou temporellement par exemple. Il est difficile de mettre l'art dans des cases.

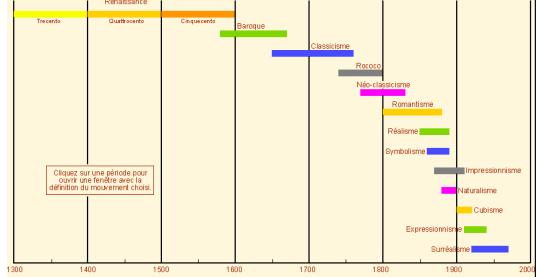


FIGURE 16: Chronologie des mouvements de peinture en Europe

## 7. Remerciements

Nous souhaitons chaleureusement remercier Benjamin Negrevergne, professeur au laboratoire LAMSADE de l'Université Paris Dauphine qui a répondu à nos questions sur son travail[6] et nous a permis de démarrer notre projet.

## Références

- [1] Keras applications, apr 2018. <https://keras.io/applications>.
- [2] Wikiart.org - visual art encyclopedia, mar 2018. <https://www.wikiart.org/>.
- [3] F. Chollet. Xception : Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.
- [4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [5] M. Ingram. Pedestrian detection using tensorflow and inception, oct 2016. <http://silverpond.com.au/2016/10/24/pedestrian-detection-using-tensorflow-and-inception.html>.
- [6] A. Lecoutre, B. Negrevergne, and F. Yger. Recognizing art style automatically in painting with deep learning. In *JMLR : Workshop and Conference Proceedings*, volume 80, pages 1–17, 2017.
- [7] A. Lecoutre, F. Yger, and B. Negrevergne. Rasta - recognising artistic style automatically, apr 2018. <http://www.lamsade.dauphine.fr/~bnegrevergne/webpage/software/rasta/rasta-web/src/frontend/>.
- [8] A. W. Services. Ami aws deep learning, mar 2018. <https://aws.amazon.com/fr/machine-learning/amis/>.
- [9] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka. Ceci n'est pas une pipe : A deep convolutional network for fine-art paintings classification. In *IEEE International Conference on Image Processing ICIP*, pages 3703–3707, 2016.
- [10] B. Zoph, V. Vasudevan, J. Shlens, and Q. Le. Automl for large scale image classification and object detection, nov 2017. <https://research.googleblog.com/2017/11/automl-for-large-scale-image.html>.