



UNIVERSITÉ
LAVAL

Classification féline avec auto-apprentissage

Stage en apprentissage automatique
Été 2020

Présenté à
M. Brahim Chaib-draa & M. Mathieu Pagé Fortin

Par
Jérémie Gince, 111 224 046, jérémie.gince.1@ulaval.ca

Université Laval
May 27, 2020

Abrégé

À faire...

Table des matières

1	Introduction	4
2	Méthodologie	5
2.1	Ensemble de données	5
2.2	Apprentissage de caractéristiques	7
2.3	Classification	9
3	Résultats	10
3.1	Apprentissage de caractéristiques	10
3.2	Classification	10
4	Conclusion	11
5	Concepts importants	12
5.1	Auto-apprentissage	12
5.2	<i>Fully Convolutional Network (FCN)</i>	12
5.3	Espace de couleur Lab	12
6	Annexe	13
	References	14

Liste des figures

1	Gamut	5
2	Gamut avec un binage de 10	6
3	Schéma simpliste de l'exemple de chat	7
4	Exemple de prédiction avec fonction de perte L1	7
5	Architecture du modèle colorieur	13

Liste des tableaux

1	Concepts à connaître	4
2	hyperparamètres pour la coloration de chats	8

1 Introduction

La colorisation est utilisée en auto-apprentissage pour colorier des images en noir et blanc. Ceci est très utile pour les historiens qui tentent d'avoir une version couleur de leurs recherches, pour les artistes qui tentent d'ajouter une touche d'intelligence artificielle à leurs arts, pour le cinéma afin de re-moderniser de vieux films très rapidement. La colorisation est aussi utile en auto-apprentissage afin d'apprendre des caractéristiques de certains type d'images. En effet, ici on utilisera la colorisation d'images dans le but de faire de la classification de races de chats avec un nombre restreint d'exemples étiquetés par des experts.

La stratégie sera donc de concevoir un modèle d'auto-apprentissage exécutant une coloration des données non étiquetés afin de préparer les caractéristiques associés aux espèces de chat. Ensuite, ces caractéristique seront fournit au classifieur supervisé qui sera entraîné sur les données étiquetés.

Concepts	Sections
Auto-apprentissage	5.1
<i>Fully Convolutional Network (FCN)</i>	5.2
Espace de couleur Lab	5.3

Table 1: Concepts à connaître

GitHub du code utilisé: [<https://github.com/JeremieGince/Cat-Breeds-Classification> (Gince, 2020)]

2 Méthodologie

2.1 Ensemble de données

L'ensemble de données utiliser est un sous ensemble du *dataset Cat Breeds Dataset* de *Kaggle* [(Aschleg & PetFinder, 2019)]. Il contient 51 races de chats pour un total de 126 066 images en couleur RGB. Pour la coloration 63 047 images sont utilisé pour l'entraînement et 10 000 sont utilisé pour la validation. On peut observer la distribution des couleurs à la figure 1. Les couleurs sont un ensemble de couples de valeurs (a, b) et ces couples forment le gamut du *dataset*. En effet, on prend les Q couples ayant la plus grande probabilités et on construit nos classes de couleurs avec ceux-ci. De cette façon chacun des pixels de tous les images de l'ensemble de données possède une classe.

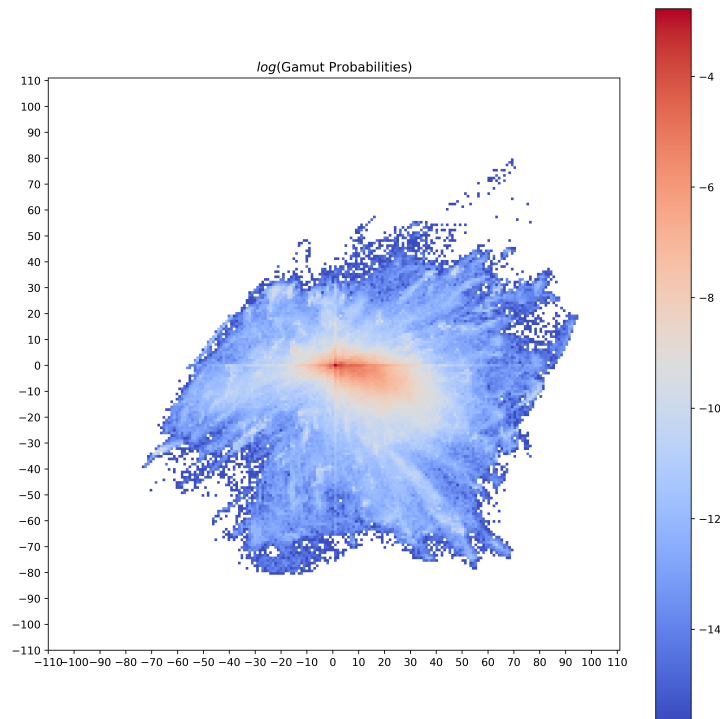


Figure 1: Gamut

Toutefois, le domaine des canaux a et b est continue et relativement grand, alors on effectue un binage sur cette densité de probabilité afin d'obtenir une discrétisation du gamut. On peut voir ce nouveau gamut biné à la figure 2.

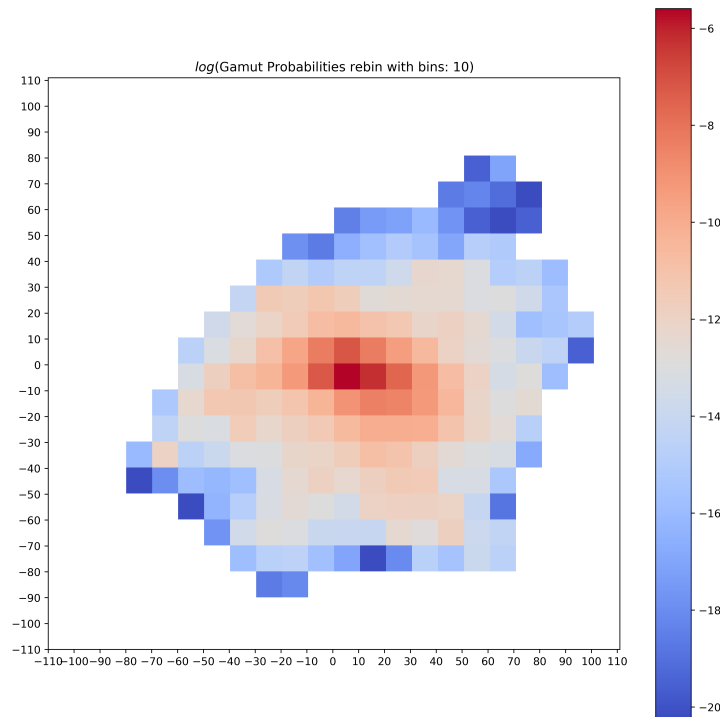


Figure 2: Gamut avec un binage de 10

Pour ce qui est de la classification, seulement 10% de l'ensemble de données est utilisé afin de simuler un ensemble de données étiqueté par des experts réduit.

À continuer...

2.2 Apprentissage de caractéristiques

La tâche ici sera de prédire la coloration des données d'entraînement transformées en images nuancées de gris. Dû à une telle tâche, l'architecture utilisée sera une *Fully Convolutional Network (FCN)* qui est un ensemble de couches de convolution avec une déconvolution en sortie afin de prédire un tenseur représentant l'image en couleur. Afin d'imager cette stratégie, on peut observer un schéma simpliste du modèle actuel à la figure 3.

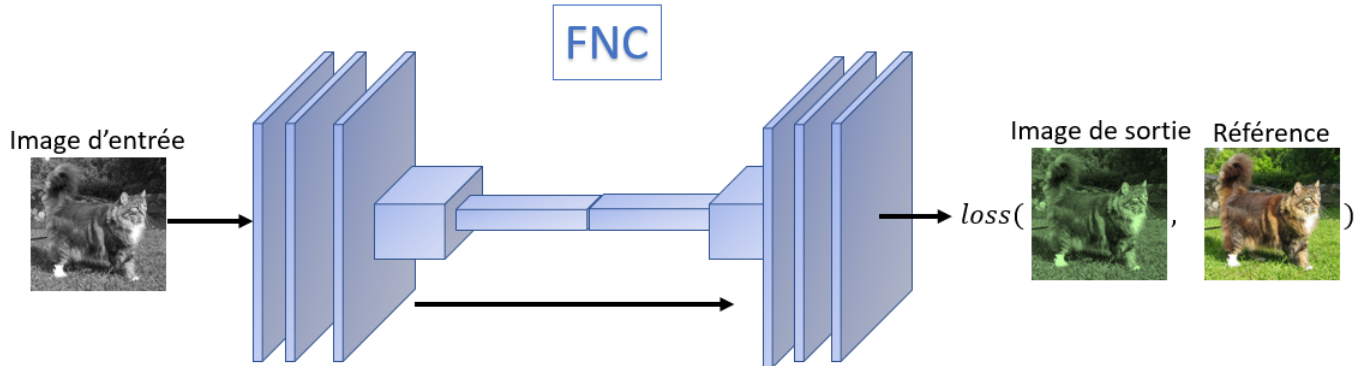


Figure 3: Schéma simpliste de l'exemple de chat

Plus précisément, l'architecture utilisée est très inspirée de [(Federico Baldassarre, 2017)] qui suggère un *FCN* utilisant *Inception-ResNet-v2* entraîné sur *ImageNet* comme extracteur de caractéristiques afin d'accélérer l'entraînement du colorier. En résumé, ce modèle prend en entrée les canaux L d'un image dans l'espace de couleur Lab et prédit les canaux ab de cette dernière. Toutefois, la fonction de perte utilisée pour une telle prédiction doit être l'erreur moyenne absolu (L1) ou l'erreur moyenne au carré (L2) et ces deux dernières fonctions semblent forcer le modèle à prédire la couleur moyenne de l'ensemble des données. Cette méthode a tout de même été testée et on peut observer le résultat sur trois exemples de prédiction à la figure 4.

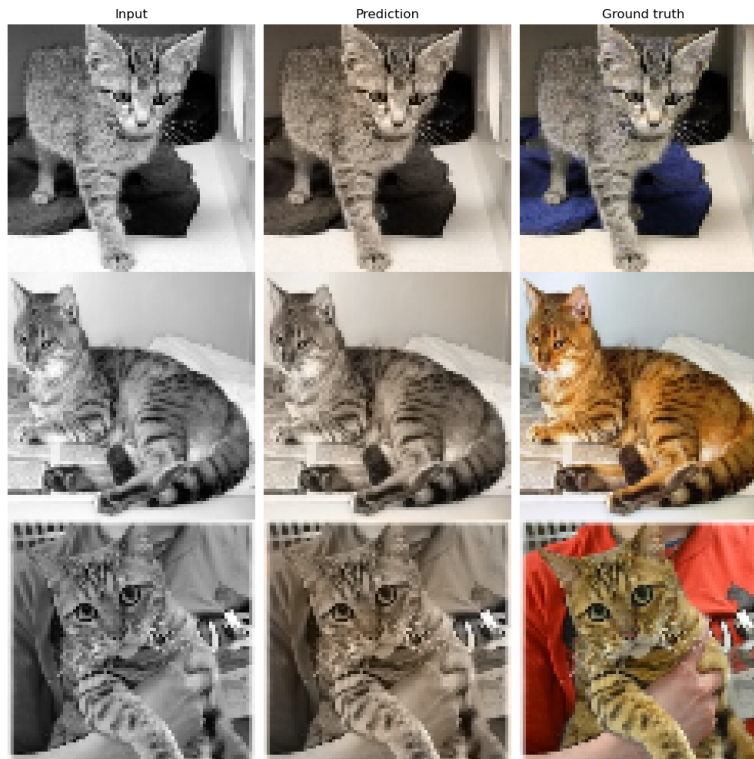


Figure 4: Exemple de prédiction avec fonction de perte L1

Afin de palier à cette mauvaise convergence, la fin de l'architecture a été modifiée avec d'avoir une sortie respectant l'idée de [(Zhang, Isola, & Efros, 2016)] qui suggère de changer la sortie afin de changer la régression pour une classification de couleur. En effet, au lieu de demander au modèle de prédire une matrice pour le canaux a et une matrice pour le canaux b , on demande au modèle de prédire une matrice de classes de couleur, aussi appelé gamut (<https://fr.wikipedia.org/wiki/Gamut>), sous forme de *one hot vector* afin de pouvoir utiliser la fonction de perte *categorical crossentropy*. L'architecture finale utilisée est donc celle présentée à l'annexe à la figure 5.

Le modèle colorier de chats a été entraîné avec les hyper-paramètres affichés au tableau 2.

Largeur d'images [<i>pix</i>]	Hauteur d'images [<i>pix</i>]	mini-batch [—]	taux d'apprentissage [—]	nombre d'époch [—]	Fonction de coût [—]	taille de gamut [—]
80	80	100	1e-3	31	mae (L1)	100

Table 2: hyperparamètres pour la coloration de chats

À mettre à jour...

2.3 Classification

Pour ce qui est de l'architecture pour la classification, elle sera semblable à celle pour la coloration. En effet, on garde les deux branches d'entrées (*encodeur* et *Inception-ResNet-v2*) ainsi que la fusion entre les deux pour finalement seulement retirer le décodeur du colorieur. Le décodeur est donc remplacé par un classifieur classique. De plus, l'entrée de l'encodeur du colorieur avait un seulement un canaux d'entrée (L), mais le classificateur recevra des images avec les trois canaux (Lab). Comme vue dans l'article [(Larsson, Maire, & Shakhnarovich, 2017)] les poids de la couche d'entrée est donc copié pour les trois canaux et divisé en trois ce qui distribue uniformément les poids pour tous les canaux et ainsi donner la possibilité au modèle d'extraire les caractéristiques de l'images en trois canaux au lieu d'un seule tout en conservant son apprentissage.

À continuer une fois que le colorieur est stable...

3 Résultats

3.1 Apprentissage de caractéristiques

À faire...

3.2 Classification

À faire...

4 Conclusion

À faire...

5 Concepts importants

5.1 Auto-apprentissage

L'apprentissage non supervisé est un type d'apprentissage où les données d'entraînement ne sont pas étiquetées. En effet, les données d'apprentissage sont fournies au modèle sans que celui-ci ait la capacité de directement savoir s'il a tort ou raison. Il doit apprendre par lui-même la corrélation entre les données sans support d'un expert. Un bon exemple de ce type d'apprentissage est le partitionnement de données ou plus communément appelé «clustering». En effet, le partitionnement de données se fait de façon non supervisée. On demande au modèle de trouver lui-même la structure cachée des données afin de regrouper les données dans des sous-groupes possédant les mêmes caractéristiques.

L'apprentissage non-supervisé est divisé en plusieurs domaines. Un de ceux-ci est l'auto-apprentissage ou *self-learning*. En fait, la particularité de ce sous-domaine de l'apprentissage non-supervisé est que ce type d'entraînement apprend lui-même à générer des étiquettes aux données d'entraînement. Il va de soit que l'absence de d'étiquettes générés par des expert rend la tâche d'entraînement plus complexe, alors il est normale de se demander qu'elles sont les avantages de ce type d'entraînement.

Dans un premier temps, l'apprentissage supervisé possède le grand désavantage d'être excessivement long à préparer. En effet, afin d'apprendre des tâches complexe, un modèle doit avoir accès à des milliers voir même des millions de données d'entraînement et il ne manque pas d'argument pour mettre en évidence la tâche colossal que représente l'étiquetage d'un aussi grand nombre de données. L'auto-apprentissage ne requiert aucun expert et donc va dans le même de ce pourquoi l'intelligence est inventé, soit l'accomplissement d'une tâche fait par un ordinateur sans intervention humaine.

Dans un deuxième temps, il existe une technique en apprentissage profond servant à transférer l'apprentissage d'un modèle à un autre modèle. Cette technique se nomme trivialement le transfert d'apprentissage. Cette dernière est beaucoup utiliser puisqu'elle permet à un modèle de converger excessivement plus rapidement vers une solution stable que si ce modèle avait commencer son apprentissage de zéro. En effet, cela est possible puisqu'il existe certaine caractéristiques récurrente à plusieurs problèmes. Prenons comme exemple la vision par ordinateur, un modèle entraîné à reconnaître des chatons aura appris à reconnaître certaine caractéristique d'un visage, d'un mammifère, de textures, d'une image etc. Ces caractéristiques peuvent logiquement s'appliquer à la reconnaissance faciale, ou la classification de types de tigre. L'auto-apprentissage sera donc très utile pour faire du transfert d'apprentissage. En effet, il est possible d'entraîner un modèle afin qu'il exécute une tâche x de façon non supervisé afin de transférer cette apprentissage à des modèles superviser entraîné à exécuter les tâches y_k .

L'auto apprentissage est actuellement utilisé pour le traitement du langage naturelle ainsi que pour la classification et obtient de bons résultats dans ces domaines. Il est moins utilisé en vision numérique actuellement puisque certaines tâches sont définitivement très complexes et requiert d'avantage de recherche dans de domaine afin d'obtenir des résultats aussi bon qu'en sémantique. Toutefois, l'auto-apprentissage promet d'être le future de l'apprentissage machine et d'être l'avenir de domaines tels que la conduite autonome, la prédiction de vidéo, la résolution d'équations différentielles complexes et bien d'autres.

5.2 Fully Convolutional Network (FCN)

À faire...

5.3 Espace de couleur Lab

À faire...

6 Annexe

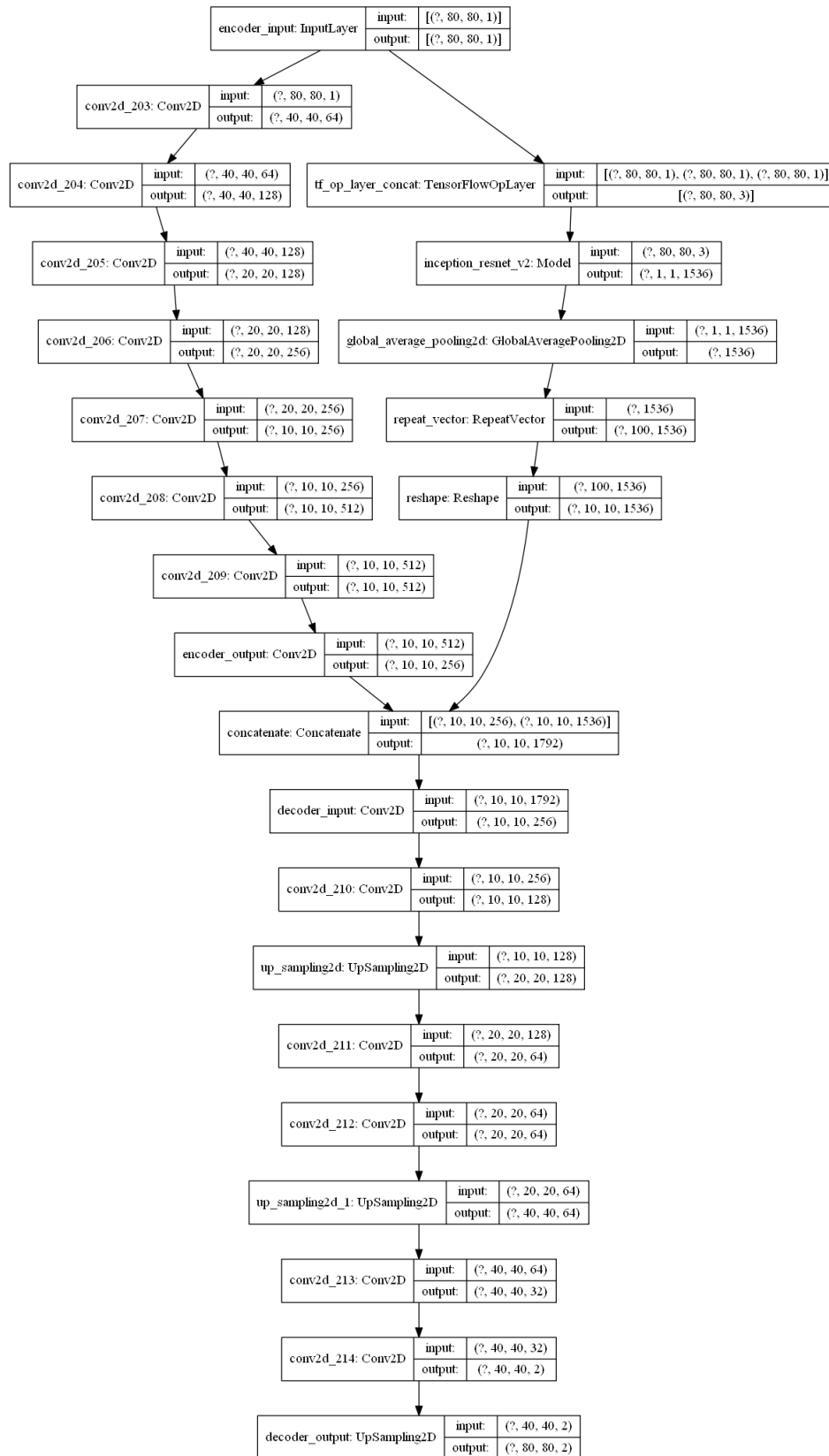


Figure 5: Architecture du modèle colorieur

References

- Aschleg, & PetFinder. (2019, Dec). *Cat breeds dataset*. Retrieved from <https://www.kaggle.com/ma7555/cat-breeds-dataset>
- Federico Baldassarre, L. R.-G., Diego Gonzalez-Morin. (2017, December). Deep-koalarization: Image colorization using cnns and inception-resnet-v2. *ArXiv:1712.03400*. Retrieved from <https://arxiv.org/abs/1712.03400>
- Gince, J. (2020, May). Cat-breeds-classification. Retrieved from <https://github.com/JeremieGince/Cat-Breeds-Classification/blob/master/Rapport.pdf>
- Larsson, G., Maire, M., & Shakhnarovich, G. (2017). Colorization as a proxy task for visual understanding. In *Cvpr*.
- Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *Eccv*.