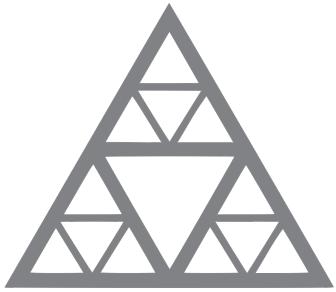


ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES



École des Ponts
ParisTech



WARNER BROS.
INTERACTIVE ENTERTAINMENT

Les Comédies Françaises sont-elles toutes bleues ?

Alex Fauduet, Virginie Loison, Maxime Poli, Nicolas Schlegel
Lise Regnier (Warner Bros. Entertainment)
Matthieu Aubry (IMAGINE - Tuteur école)

Rapport de projet de département IMI

Second Semestre 2020

Remerciements

Avant tout, nous souhaitons remercier l'ensemble des personnes qui nous permis de réaliser ce projet dans les meilleures conditions possibles.

Merci à Manon Baudel et à Lise Regnier pour nous avoir proposé ce sujet très intéressant, grâce auquel nous avons beaucoup appris.

Merci à Stephano Perasso pour nous avoir permis de lancer ce projet.

Merci à Lise Regnier pour nous avoir suivis tout au long du projet. Il fût très agréable de la rencontrer chaque semaine, elle, son savoir-faire et sa bonne humeur communicative.

Merci également à Manon Baudel pour s'être assurée que notre projet se déroulait dans les meilleures conditions.

Enfin, merci à Mathieu Aubry qui grâce à son expertise, a su nous réorienter et répondre à nos interrogations.

Introduction

Cadre du projet

Dans le cadre de notre projet de département IMI, nous avons travaillé avec Warner Bros pour mettre au point un outil de classification d'affiches de films. Le but était de pouvoir prédire les différents genres d'un film à partir de son affiche. Pour ce projet, nous étions accompagnés par Lise Regnier, *data scientist* chez Warner Bros.

Ce projet est motivé par le constat que des affiches d'un même genre ont souvent la même apparence. Un exemple bien connu est celui des affiches des comédies françaises, qui semblent avoir adopté un fond bleu, un grand titre dans les tons jaune, et les stars du film en gros plan comme le montre l'exemple ci-dessous.



FIGURE 1 – Affiches de comédies françaises

Ce projet peut avoir des applications diverses. Il pourra par exemple être utile pour le choix d'une affiche d'un futur film par le département marketing de Warner Bros France.

Redéfinition du cadre de l'étude

Le fil rouge de notre projet a été le suivant : construire un algorithme qui puisse prédire le genre associé à un poster avec le meilleur taux de réussite possible. C'est ce sur quoi nous avons principalement travaillé. Notre solution finale repose sur des réseaux de neurones.

En échangeant avec notre client, nous avons compris qu'il était également important de pouvoir contextualiser le résultat renvoyé par un algorithme, notamment en donnant des éléments visuels qui donnent des intuitions sur la prédiction faite. Par exemple, il est pertinent pour notre client d'afficher les posters que l'algorithme considère comme proches d'un poster test. Nous avons gardé cette idée en tête au long de notre travail. C'est pourquoi nous avons conservé, en complément de notre réseau final, des méthodes ayant une *accuracy* moins bonne, mais qui contextualisent le résultat renvoyé.

D'un point de vue plus technique, le problème qui nous a été posé par le client est un problème de classification *multi-label*. Une meilleure approche est de commencer en formulant le problème en *mono-label*, puis de l'adapter au *multi-label*. En effet, les algorithmes mono-label sont facilement adaptables en *multi-label*, et il est illusoire de faire du *multi-label* si on n'a pas déjà des algorithmes performants en *mono-label*.

Structure du rapport

Lise Regnier, notre client chez Warner, nous a donné accès à la base de donnée payante d'Allociné. Cette base de donnée est complète mais très déséquilibrée, les genres des films n'étant pas répartis uniformément. Nous parlerons dans ce rapport du traitement que nous avons fait pour utiliser au mieux cette base de données dans nos méthodes de *machine learning*.

Nous avons exploré plusieurs méthodes. Dans ce rapport, nous parlerons uniquement de celles qui ont donné de bonnes performances, ou qui nous ont permis d'avancer.

Nous commençons par deux préliminaires : un lexique, et une synthèse de l'état de l'art. Dans une première partie, nous présentons la base de données et les manipulations que nous lui avons appliquées en raison des problèmes de déséquilibre entre les différentes classes. Dans un second temps, nous parlons des résultats que nous avons obtenus en utilisant des algorithmes de type "plus proche voisin". Enfin, nous exposons les résultats obtenus en utilisant des réseaux de neurones profonds. Nous synthétisons ensuite les avantages et inconvénients des méthodes que nous avons utilisées, ainsi que l'utilisation directe que Warner Bros pourra en faire. Nous les comparons également à l'état de l'art.

Table des matières

Lexique	5
État de l'art	6
1 La base de données	7
1.1 Présentation de la base de donnée	7
1.2 Équilibrage de la base de données	7
1.3 Constitution des ensembles d'entraînement et de test	8
2 Méthode des plus proches voisins	9
3 Réseaux de neurones profonds : approche directe	11
4 Réseaux de neurones profonds : <i>Transfer learning</i>	13
4.1 Plus proches voisins	14
4.2 Transfert entre réseaux	15
4.3 Arbre de décision	22
Conclusion	24
Réponse aux besoins métiers du client	24
Pistes d'amélioration	25

Lexique

Classification : Méthode qui consiste à associer une classe parmi un ensemble de catégories à l'objet que l'on doit classer. On distingue la classification *mono-label* de la classification *multi-label*. Dans le premier cas, on n'attribue qu'une seule classe à l'objet que l'on doit classer, dans le deuxième, on peut lui attribuer plusieurs classes.

K-plus proches voisins (k-NN) : Méthode d'apprentissage supervisé non paramétrique qui consiste à représenter les variables explicatives des éléments de l'ensemble d'entraînement comme des points d'un espace vectoriel normé.

La prédiction d'un élément se base sur ses k plus proches voisins dans cet espace vectoriel normé. Dans un problème de classification, la classe attribuée à l'élément est la classe majoritaire parmi les k voisins.

Lorsque l'on met au point un k-NN, on doit se poser deux questions fondamentales : la manière dont on représente les données et la métrique que l'on utilise.

Transfer Learning : Pratique qui vise à transférer des connaissances d'une ou plusieurs tâches sources vers une ou plusieurs tâches cibles. Elle est particulièrement utile lorsqu'on dispose d'une base de données de petite taille : on récupère alors un modèle entraîné sur une base de données de grande taille qu'on va utiliser pour notre nouvelle tâche plus spécifique. On ne repart pas de zéro, et on utilise les connaissances acquises précédemment.

Arbre de décision : Méthode permettant de représenter un ensemble de choix sous la forme d'un arbre. En apprentissage supervisé, on peut construire un arbre de décision à partir des données d'entraînement et s'en servir pour faire de la prédiction.

Forêt d'arbres de décision (*Random Forest*) : Pour apporter plus de robustesse à la méthode des arbres de décision, les méthodes de type *random forest* consistent à entraîner séparément plusieurs arbres de décision sur des sous-ensembles de l'ensemble d'entraînement. La prédiction se fait ensuite par vote majoritaire entre les différents arbres.

ResNet : Les réseaux de type ResNet sont des réseaux de neurones convolutionnels destinés à la classification d'images. Ce sont des réseaux de référence. Cette méthode utilise des "blocs résiduels" qui permettent aux connections de sauter des couches de convolution. Cette méthode a permis de développer des réseaux de neurones plus profonds et plus performants.

ImageNet : Base de données d'images. Elle contient plus de dix millions d'images annotées.

Précision d'un classifieur *mono-label* : Moyen d'évaluer la performance d'un outil de classification *mono-label*. Il correspond au pourcentage de bonnes réponses données sur un ensemble. On utilise aussi l'anglicisme "*accuracy*".

Matrice de confusion d'un classifieur : Pour comprendre plus précisément un classifieur, on représente ses performances dans des matrices. Le terme (i, j) de la matrice contient la proportion d'éléments de la i -ème classe qui ont été classés comme dans la j -ème classe.

État de l'art

D'autres personnes se sont déjà penchées sur la classification de films selon leur genre. Il existe de la bibliographie sur la classification de genres se basant sur les posters, sur les bandes-annonces, ainsi que sur les synopsis de films. Citons quelques sources qui nous ont particulièrement inspirés.

Certains articles utilisent des méthodes purement statistiques sur des *features* "bas niveau" des posters. C'est par exemple le cas de la référence [6]. Cet article utilise des histogrammes de couleurs sur des zones délimitées des posters. Plusieurs approches sont testées, qui sont des combinaisons d'approches Bayésiennes et d'algorithme k-NN (k-Nearest Neighbors). Cet article développe aussi beaucoup l'importance du pré-traitement des données, et montre son impact sur la performance des algorithmes.

Certains utilisent directement des algorithmes ayant fait leur preuves pour la classification d'images, sans tenir compte des spécificités de ce problème. On trouve ici [2] un exemple d'implémentation. Les posters sont traités directement par un réseau de neurones convolutif (CNN).

On peut enfin trouver des idées d'algorithmes de *deep learning* qui combinent des *features* "bas niveau" et des *features* "haut niveau". Citons par exemple l'article [1]. Cet article utilise en parallèle un CNN et un modèle de reconnaissance d'objet : YOLO [7].

Enfin, des méthodes type *transfer learning* se basent sur des réseaux déjà entraînés sur des bases de données de classification d'image, et les adaptent pour le problème de classification de posters. L'article [3] applique cette méthode sur une base de données constituée de posters turcs.

Ces sources utilisent des méthodes de complexités variables, mais ont des performances similaires. Aucune ne prédit avec justesse le genre d'un poster plus d'une fois sur 2.

1 La base de données

1.1 Présentation de la base de donnée

Pour ce projet, Warner Bros. nous a donné accès à la base de données de posters de films d'Allociné. Cette base de donnée contenait à l'origine 17513 films et diverses informations comme leur titre français, leur titre original, leurs genres, leur synopsis, leurs acteurs... Pour notre projet, nous nous sommes simplement concentrés sur leurs affiches et leurs genres.

Enfin, chaque film est associé à un, deux voire trois genres parmi 36 genres.



FIGURE 2 – Posters pris aléatoirement dans la base de données avec leurs genres

1.2 Équilibrage de la base de données

Comme le montre le tableau de répartition des genres, la distribution des genres est très déséquilibrée.

	Genre 1	Genre 2	Genre 3		Genre 1	Genre 2	Genre 3
Action	1016	615	143	Espionnage	31	45	11
Animation	986	44	15	Expérimental	12	6	3
Arts Martiaux	32	17	8	Famille	100	326	183
Aventure	478	547	153	Fantastique	333	347	152
Biopic	251	205	51	Guerre	143	154	72
Bollywood	8	21	8	Historique	147	242	90
Classique	1	0	0	Judiciaire	10	42	18
Comédie	3004	705	216	Movie night	0	1	0
Comédie dramatique	1278	97	4	Musical	81	179	73
Comédie musicale	115	54	16	Opera	29	1	0
Concert	8	3	0	Policier	581	386	112
Dessin animé	2	0	0	Péplum	13	3	4
Divers	255	0	0	Romance	252	940	316
Documentaire	1619	46	5	Science fiction	328	225	105
Drama	0	0	1	Show	1	1	0
Drame	4648	1442	369	Sport event	5	13	3
Epouvante-horreur	584	245	74	Thriller	947	848	323
Erotique	18	32	12	Western	163	34	16

TABLE 1 – Attribution des différents genres aux films dans la base de données initiale

Tout d'abord, pour nous ramener à un problème de classification *mono-label*, nous avons décidé de ne considérer que le genre principal de chaque film.

Cependant, certains genres restaient trop peu représentés pour pouvoir être correctement prédicts par des algorithmes de *machine learning*. Ainsi, nous avons décidé de procéder à des rassemblements de classes pour équilibrer notre base de données. Les films furent répartis dans 7 catégories :

- La catégorie "Action" qui réunit les films d'action, de guerre et d'arts martiaux,
- La catégorie "Animation" qui contient les films d'animation et les dessins animés,
- La catégorie "Comédie",
- La catégorie "Comédie Dramatique",
- La catégorie "Documentaire",
- La catégorie "Drame",
- La catégorie "Thriller-Policier" qui rassemble les thrillers, les films policier ainsi que les films d'espionnages et judiciaires.
- Les genres "Classique", "Concert", "Comédie musicale", "Aventure3", "Opéra", "Famille", "Show", "Divers", "Erotique", "Sport Event", "Expérimental" et "Movie night" sont ignorés car trop peu représentés, ou trop peu porteurs de sens.

Notons que lorsque le premier genre d'un film appartenait à l'une des catégories que nous avions décidé de garder, on lui attribuait ce genre. Lorsque ce n'était pas le cas, on faisait de même avec son deuxième genre puis troisième genre. Les films dont aucun genre n'appartenait aux genres que nous avions décidé de garder étaient ignorés.

À la fin de l'équilibrage de la base de données, la base de donnée contient 11935 films, dont voici la répartition des genres :

Genre	Nombre de films
Action	885
Animation	710
Comédie	2332
Comédie dramatique	1086
Documentaire	1401
Drame	4225
Thriller-Policier	1296

TABLE 2 – Attribution des différents genres aux films dans la base de données équilibrée

1.3 Constitution des ensembles d'entraînement et de test

Pour avoir une base de donnée équilibrée, nous avons décidé de prendre 710 films dans chaque catégorie parmi lesquels, 595 films étaient utilisés dans les données d'entraînement et 105 dans les données de test. Au total, nous avons donc entraîné nos modèles sur 4900 films différents dont 4165 pour l'entraînement et 735 pour le test.

Cette réduction du nombre de film fût nécessaire pour nous affranchir de certaines tendances des algorithmes de *machine learning* qui consistaient à prédire uniquement des comédies et des drames (les genres les plus représentés) pour maximiser leur précision.

Le nombre de films dans les ensembles d'entraînement et de test ont été limités par le nombre de films du genre le moins représenté, c'est-à-dire "Animation". En effet, nous souhaitions faire des ensembles d'entraînement et de test équirépartis en genres.

2 Méthode des plus proches voisins

Les algorithmes de la famille "k-plus proche voisins" (k-NN) sont très utilisés dans les problèmes de classification. Ils sont en effet facilement implémentables, et permettent une bonne visualisation. Des algorithmes de k-NN ont déjà été utilisés sur le problème de classification d'un poster de film, par exemple dans [6]. Nous allons présenter et analyser les premiers résultats que nous avons obtenus avec ce type d'algorithme.

Notre toute première idée a été de représenter les posters grâce à l'implémentation naturelle de leurs images. Si on prend une image RGB de largeur w et de hauteur h , elle est représentée grâce à une matrice 3D de taille $(w, h, 3)$. Nous avons redimensionné cette matrice pour avoir un vecteur de dimension $w \times h \times 3$. Concernant la métrique, nous avons utilisé la distance L^2 .

Nous attendions peu de performance de la part de cet algorithme. Il est en effet très naïf. Les *features* utilisées ici sont très lourdes et peu porteuses de sens. Cette intuition a été confirmée par l'expérience.

La figure 3, permet de quantifier la performance de cet algorithme. La première courbe, tracée en orange, est l'*accuracy* moyenne d'un k-NN sur l'ensemble de test. La deuxième, tracée en bleu, est une valeur de référence pour comparaison. C'est l'*accuracy* moyenne d'un classifieur qui assigne un genre au hasard à chaque poster qu'il rencontre. Comme l'ensemble de test est équiréparti sur les 7 genres, cette *accuracy* moyenne de référence est de $\frac{1}{7} \approx 14\%$.

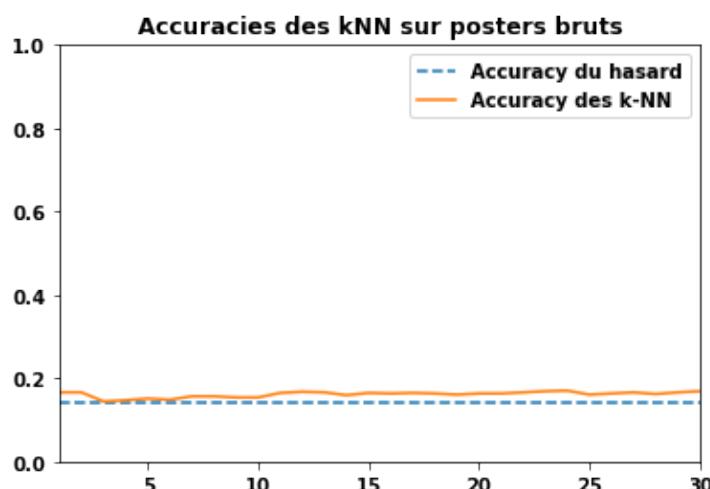


FIGURE 3 – *Accuracies* du k-NN sur posters bruts sur l'ensemble de test

Cette courbe montre que les k-NN sur posters bruts sont aussi performants que le hasard. Ce qui n'est évidemment pas satisfaisant.

Analysons maintenant sur quelques exemples les plus proches voisins retournés par le k-NN. Cela nous donne une idée des caractéristiques auxquelles il accorde de l'importance.

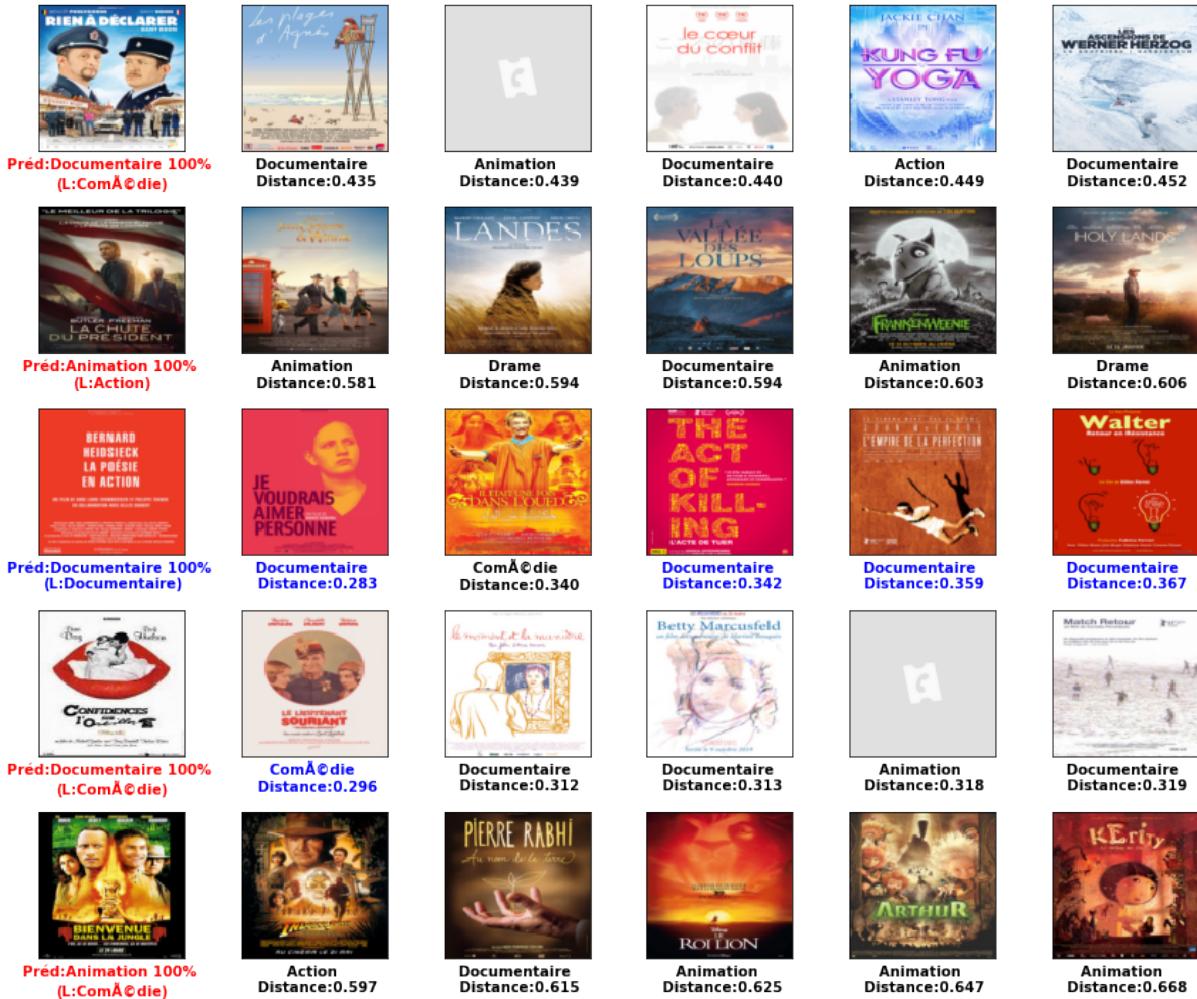


FIGURE 4 – Exemples de plus proches voisins, k-NN sur posters bruts

Les posters sur la colonne de gauche sont des posters tirés aléatoirement dans l'ensemble de test. Les posters sur les 4 autres colonnes sont leurs plus proches voisins dans l'ensemble d'entraînement. Pour chacun des plus proches voisins, on affiche son genre, et la distance qui le sépare du poster test.

On voit qu'avec les posters bruts comme *features*, le k-NN se base sur la colorimétrie pour prendre sa décision. Cette information n'est manifestement pas suffisante. Par exemple, sur la première ligne, le poster test a un ciel dégagé en fond. Ses plus proches voisins sont donc très clairs, et plutôt bleutés. Mais la présence d'un ciel clair n'est pas caractéristique d'un genre.

Cette idée a été confortée par les résultats que nous avons eu en utilisant d'autres *features* basés sur la colorimétrie. Nous avons calculé les histogrammes de couleurs RGB et LAB de chaque poster, et avons fait des k-NN dessus. Les résultats que nous avons obtenus étaient aussi peu probants, pour la même raison : la colorimétrie seule d'un poster ne suffit pas pour prédire son genre.

Cette première approche naïve nous a apporté deux choses :

- La certitude qu'il faudra utiliser des *features* complexes pour traiter notre problème, qui ne sont pas uniquement basés sur la colorimétrie.
- L'implémentation d'un algorithme k-NN qui nous sera utile pour la suite.

3 Réseaux de neurones profonds : approche directe

Une limitation importante de la méthode précédente est qu'elle se contente d'informations "bas niveaux" : elle ne fait que comparer des caractéristiques simples des posters, comme la répartition des couleurs.

Or, il est logique de penser que des éléments plus "haut niveau" sont nécessaires pour caractériser le genre d'un film : la présence de personnages, d'objets, leur nombre et position dans le poster sont ce sur quoi les humains se reposent en majorité pour effectuer cette tâche.

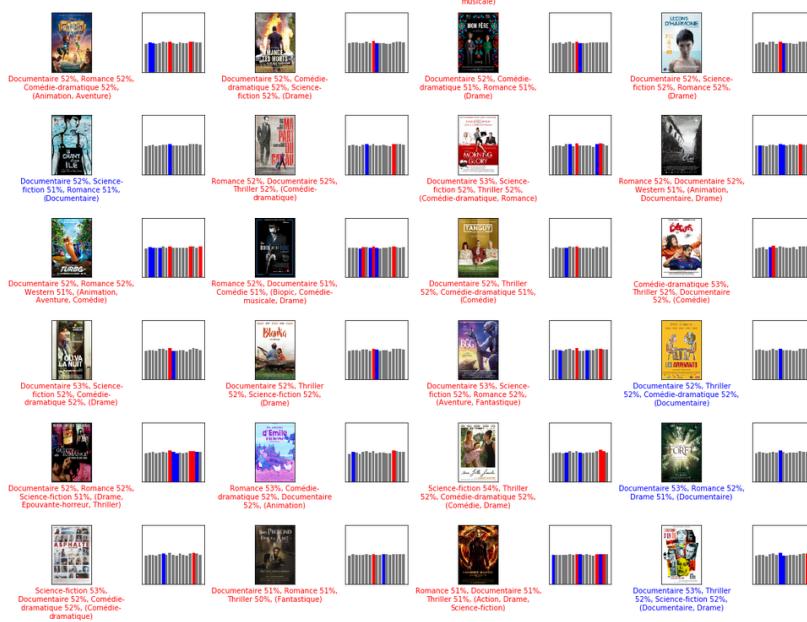
La solution moderne à ce genre de problèmes est l'utilisation de réseaux de neurones artificiels. On ne comprend que très peu les critères qu'ils utilisent pour comprendre et interpréter l'information contenue dans une image. Ils sont pourtant très efficaces, à condition d'être suffisamment entraînés.

Notre première idée a été de construire notre propre réseau et de l'entraîner sur nos données "*from scratch*". Cette idée a déjà été implémentée et a donné de bons résultats : [2] [1]. Nous avons implanté le même type de réseau que dans la référence : un CNN (pour *Convolutional Neural Network*). Ces réseaux sont en effet des réseaux de référence pour l'apprentissage *from scratch*. Nous avons utilisé la même structure que celle décrite en [2].

Cette approche a vite posé des problèmes techniques. En effet, pour qu'un réseau neuronal entraîné en démarrant à zéro soit performant, il faut que la base de donnée sur laquelle il est entraîné soit de bonne qualité. Elle doit être très fournie (beaucoup de films), et chaque genre doit y être bien représenté.

La taille de notre base de données était trop modeste, c'est pourquoi nos réseaux neuro-naux n'arrivaient pas à apprendre dessus. En pratique, pour entraîner des réseaux performants sur des problèmes similaires, on a besoin de quelques dizaines de milliers d'images, contre seulement 4000 dans notre ensemble d'entraînement après les pré-traitements. À cause de cela, nos réseaux faisaient systématiquement la même prédiction.

Lorsque le réseau neuronal était entraîné sur une base de donnée déséquilibrée, il prédisait systématiquement le genre le plus représenté (voir 5b). En revanche, quand le réseau neuronal était entraîné sur une base de donnée équilibrée, il prédisait tous les genres possibles avec quasiment la même probabilité (voir 5a).



(a) Entraînement sur un ensemble d'entraînement équiréparti



(b) Entraînement sur un ensemble d'entraînement non équiréparti

FIGURE 5 – Prédiction sur quelques posters tests d'un CNN entraîné sur notre base de données

Nous avons songé à utiliser les mêmes bases de données que [2] et [1], pour espérer

avoir de meilleurs résultats. Mais cela n'aurait pas été pertinent. En effet, Lise Regnier, notre client chez Warner France, nous a expliqué que les films avaient une affiche différente par pays, car chaque pays a ses propres conventions d'affiches. Or, les bases de données de [2] et [1] sont des affiches américaines de films. Elles ne sont donc pas adaptées à notre projet, basé sur des affiches françaises.

Les caractéristiques de notre base de donnée ne nous permettent pas de faire une approche directe, c'est-à-dire d'entraîner un réseau neuronal directement sur notre base de donnée. Il nous a donc fallu trouver une méthode plus intelligente.

4 Réseaux de neurones profonds : *Transfer learning*

Nous avons donc abandonné l'approche précédente, pour nous concentrer sur la réutilisation de réseaux déjà entraînés pour des tâches similaires. C'est ce qu'on appelle du *transfer learning* : la ré-utilisation des connaissances d'un algorithme entraîné pour une certaine tâche à un nouveau problème proche du précédent. Cette méthode est classique en classification d'image [4]. Elle a été appliquée au problème de classification de posters dans [3]. Remarquons que cet article traite un problème proche du nôtre : il est basé sur une base de donnée réduite, celle des affiches de films turcs.

Nous avons choisi d'utiliser les résultats intermédiaires d'un réseau de neurone profond dédié à la classification d'images. L'idée est qu'un tel algorithme, pour être capable de reconnaître et classifier des objets dans une image, doit pouvoir en reconnaître des éléments significatifs qui vont au-delà de la luminosité et des couleurs (forme, texture, position...). Ainsi, si au lieu d'aller au bout de la tâche de classification, on arrête l'un de ces réseaux à une étape suffisamment avancée, il est très probable qu'il ait trouvé les caractéristiques essentielles du contenu de l'image. Cette approche présente l'inconvénient majeur qu'il est très difficile de se représenter et de donner un sens à cette information, mais elle suffit à appliquer des classificateurs classiques sur des composantes plus pertinentes du poster.

Dans la suite, nous avons utilisé le réseau ResNet [5] : il s'agit de l'une des architectures les plus performantes sur la base de donnée ImageNet, qui contient une grande variété d'images à classer dans de très nombreuses catégories. En récupérant ce réseau déjà entraîné, nous pouvons donc extraire les informations les plus pertinentes.

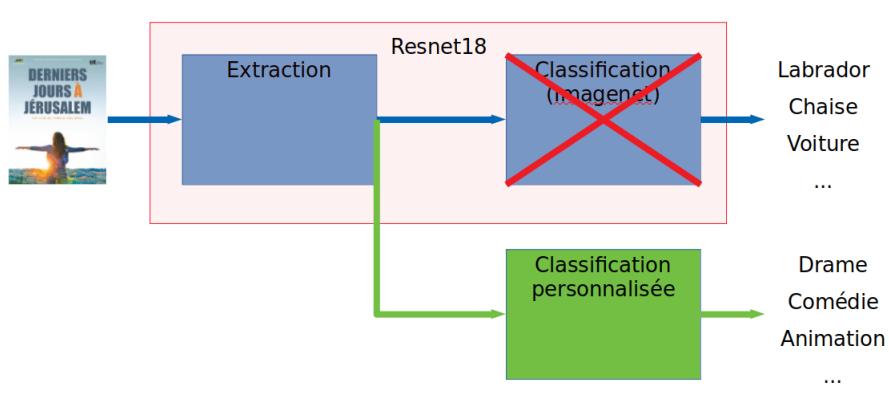


FIGURE 6 – Réutilisation de Resnet18

4.1 Plus proches voisins

A partir des résultats intermédiaires de ResNet, on peut donc réutiliser l'algorithme des plus proches voisins précédent : plutôt que d'utiliser les pixels bruts du poster afin de déterminer leur distance entre eux, nous utilisons à la place les composantes extraites par le réseau de neurones.

Cette méthode a eu deux avantages majeurs dans notre projet : en premier lieu, les résultats sont bien meilleurs que ceux de nos algorithmes précédents : nous obtenons des résultats plus de deux fois plus précis que le hasard là où on le dépassait à peine avant.

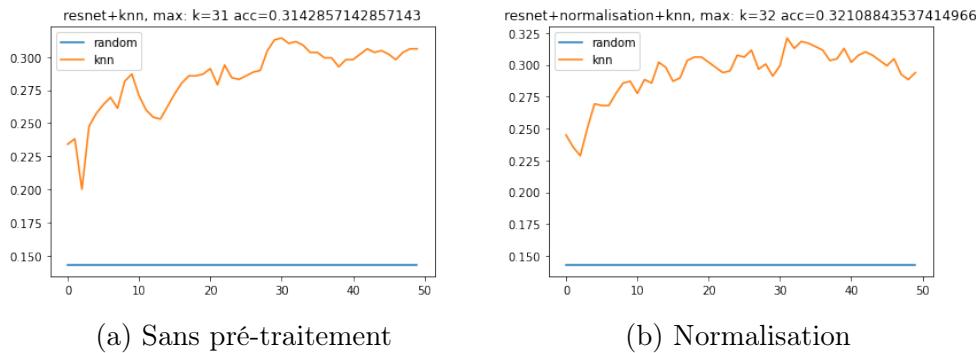


FIGURE 7 – Évolution de l'*accuracy* du kNN après différents pré-traitements sur les *features* renvoyées par ResNet

Deuxièmement, ces résultats restent interprétables : les réseaux de neurones font généralement perdre tout sens aux données au profit des résultats, mais une approche par plus proches voisins, grâce à la visualisation des voisins en question, permet de redonner un sens aux critères utilisé pour classifier.

Cette particularité tenait particulièrement à cœur à notre tutrice chez Warner Bros : si avoir de bons résultats est évidemment important, pouvoir les expliquer et les contextualiser est crucial.

L'affichage des plus proches voisins d'un film serait ici utilisable comme assistant de prise de décision pour le service marketing de Warner Bros. En effet, lorsqu'il faut créer une affiche pour un film, il est crucial que l'aspect visuel de l'affiche fasse penser à des affiches de films déjà existants, et proches du film qu'on cherche à promouvoir. Ainsi, en un coup d'oeil, le public peut facilement se donner une idée sur le film associé à l'affiche, pour décider d'aller le voir ou non.

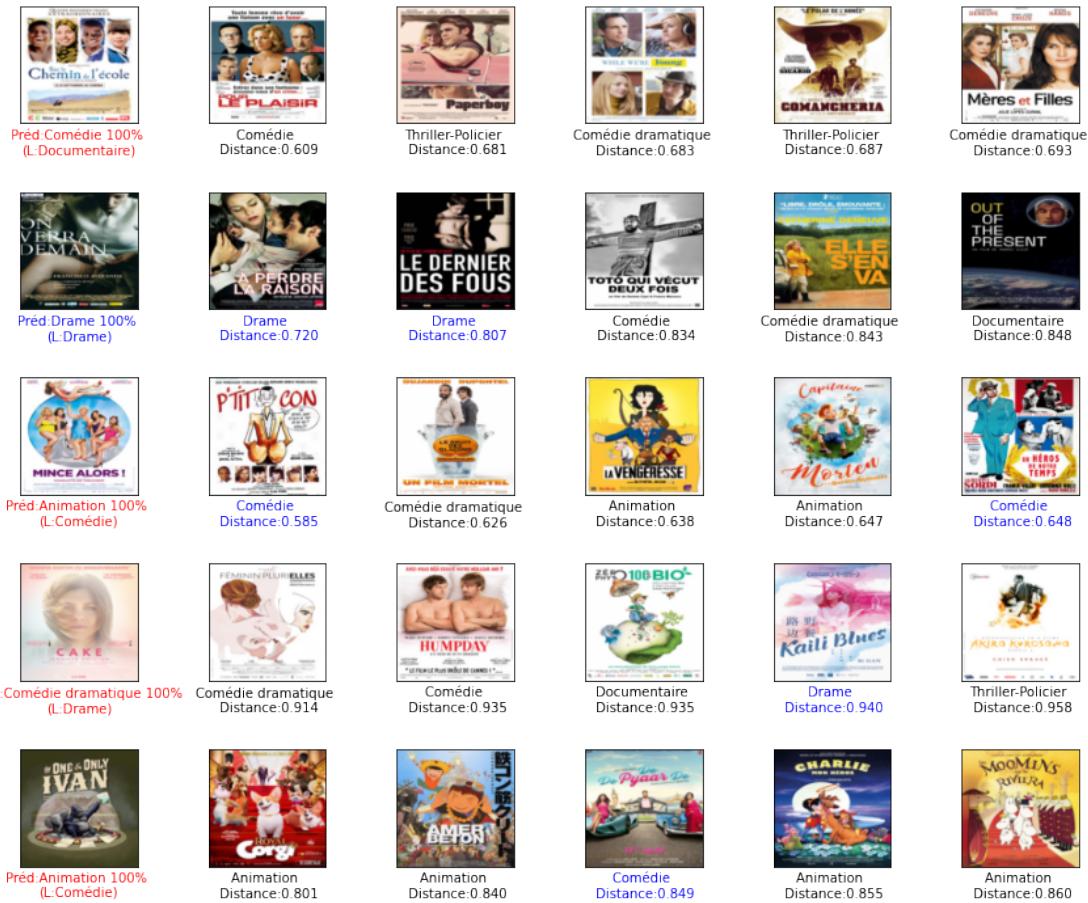


FIGURE 8 – Visualisation de quelques voisins de 5 films

Ainsi, le service marketing pourrait utiliser notre algorithme ResNet+kNN sur plusieurs candidats d'affiches pour un film. L'algorithme renverrait les affiches existantes proches des candidats. Et le service marketing pourrait alors choisir l'affiche dont les voisins sont les plus proches des idées qu'ils souhaitent promouvoir.

4.2 Transfert entre réseaux

Ici, on parlera en particulier de transfert entre réseaux de neurones : c'est une nouvelle couche de neurones qui s'occupera de la classification. On a ainsi essentiellement remplacé la dernière couche de ResNet par une autre couche similaire, plus adaptée à notre problème. Concrètement, cette nouvelle couche va effectuer une régression linéaire vers les classes de notre problème.

L'utilisation de ce nouveau réseau se fait en deux phases.

En premier lieu, il faut entraîner cette nouvelle couche : le reste du réseau a déjà été entraîné et devrait être capable d'extraire l'information dont on a besoin, mais la couche de classification ne connaît encore rien de notre problème, et il faut donc l'entraîner individuellement.

Nous ne souffrons pas ici des difficultés précédemment rencontrées par notre réseau "à partir de rien", puisque que nous n'avons qu'une couche à entraîner plutôt qu'un réseau entier, et donc une quantité moindre de données est nécessaire.

Nous avons également choisi à ce moment-là de faire de la *data augmentation*, c'est-à-dire d'appliquer des opérations différentes sur les images chaque fois qu'elles sont chargées dans le réseau, ce qui revient à augmenter artificiellement la taille de l'ensemble d'entraînement. Cette pratique est utilisée lorsque la base de donnée est un peu petite pour l'usage qu'on souhaite en faire. Nous avons tout d'abord redimensionné toutes les images pour qu'elles aient une taille 256×256 . Des tailles plus faibles donnaient de moins bons résultats, et des tailles plus grandes demandaient une durée d'entraînement qui était difficilement gérable pour nous. Nous avons également appliqué de manière aléatoire une saturation des couleurs et un ajustement de contraste ; nous avons rogné les images et effectué ou non, de manière aléatoire, une symétrie par rapport à l'axe vertical central.

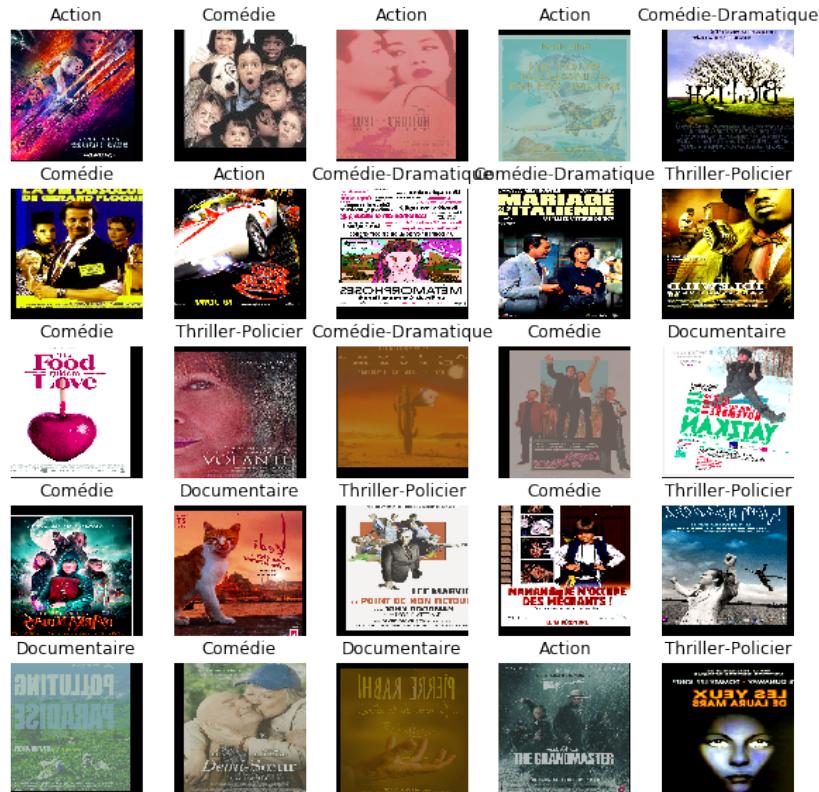


FIGURE 9 – Exemple de *batch* après *data augmentation*

À ce moment du projet nous avions conscience de l'importance de certains choix pour l'entraînement du réseau, comme celui du taux d'apprentissage ou de la taille des *batches*. Nous avons ajusté ces deux hyperparamètres en regardant les résultats après un entraînement de 30 époques. Nous avons conservé dans toute la suite une taille de *batch* de 32 et un taux d'apprentissage de 0.001 pour les quinze premières époques et de 0.0001 pour les quinze suivantes.

Après 30 époques, l'*accuracy* sur l'ensemble d'entraînement et sur l'ensemble de validation restaient constantes, on a arrêté les entraînements à ce moment-là. On obtient alors **0.431** d'*accuracy* sur l'ensemble de test, ce qui est le meilleur résultat obtenu jusqu'à présent.

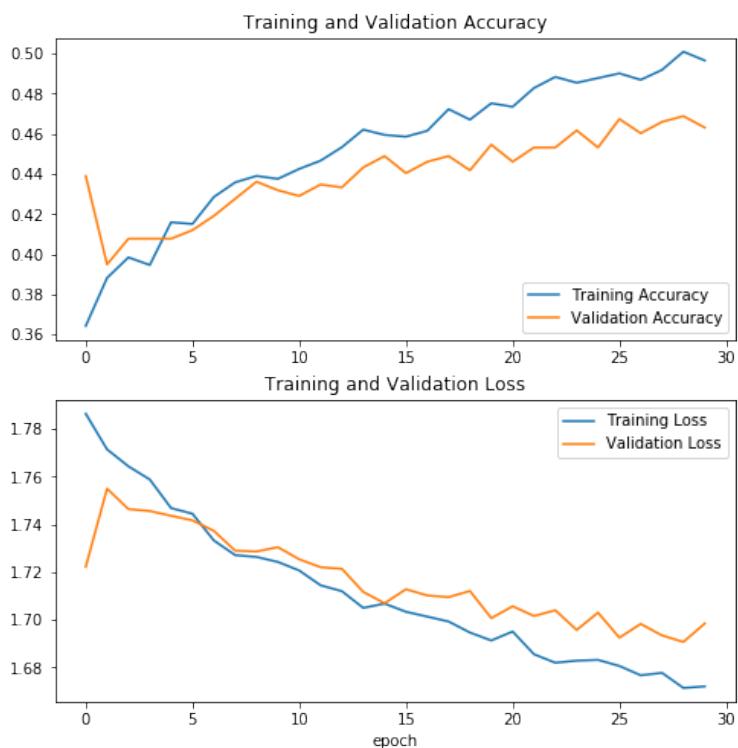


FIGURE 10 – Entraînement de la tête de classification

La deuxième partie est appelée *fine-tuning* : jusqu'à présent, on espérait que les composantes extraites par ResNet étaient pertinentes pour l'application qu'on veut en faire, mais ceci n'est en fait pas évident. Par exemple, si le ResNet initial a été entraîné pour reconnaître un chat, il ne lui est pas pertinent d'analyser le style, et il ne fera pas la différence entre une photo d'un chat réel et un dessin de chat. Ce genre d'information est pourtant très important pour nous, et permet entre autres de faire la différence entre le genre "Animation" et les autres.

Nous voulons donc améliorer la première partie du réseau, issue de ResNet. Il ne s'agit pas de tout réapprendre, puisqu'on a bien vu que l'information traitée par ResNet a fourni

une amélioration non négligeable de nos résultats. Il faut plutôt apporter de petites améliorations pour adapter une méthode déjà très performante au cas particulier auquel on veut l'appliquer. Cet entraînement se fait donc en avançant par pas plus petits que précédemment, pour ne pas trop s'éloigner des valeurs du ResNet initial.

La problématique principale a été de trouver comment dégeler la partie ResNet et l'entraîner peu à peu. En ne faisant pas attention au taux d'apprentissage par exemple, on peut très facilement sur-entraîner dans ces cas-là. L'exemple ci-dessous, montre les résultats pour un des entraînements qu'on a fait, avec peu d'époques pour la première partie, sans *data augmentation* et avec un taux d'apprentissage qui reste de 0.0001 au début du *fine-tuning* qui ne consistait qu'à dégeler la dernière couche convolutionnel du ResNet.

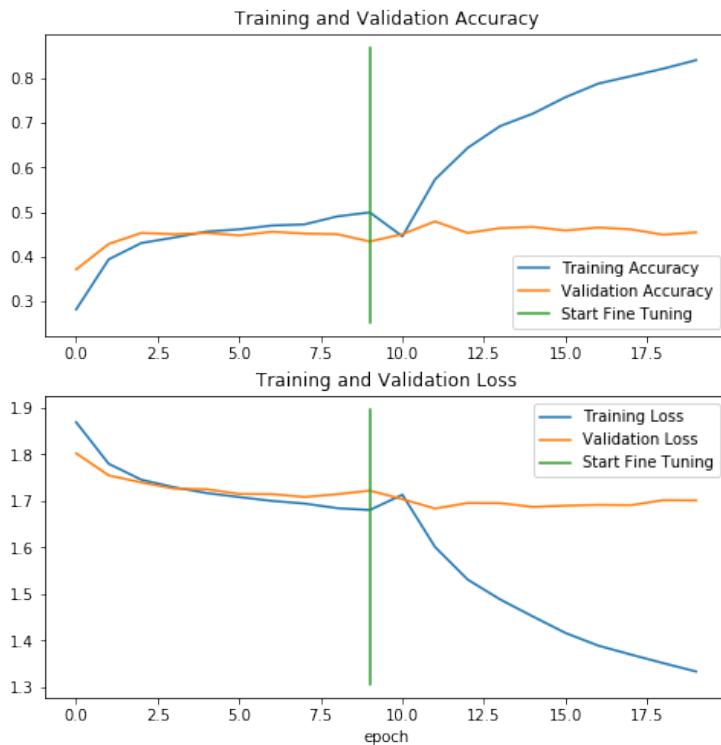


FIGURE 11 – Exemple de sur-apprentissage

On peut voir qu'avec de tels choix le réseau sur-apprend très vite et l'*accuracy* sur l'ensemble de validation reste constante : on reconnaît de l'*overfitting*. Le réseau ne sait pas mieux reconnaître le genre d'un film sur lequel il ne s'est pas entraîné, mais il a par contre ajusté ses paramètres pour très bien reconnaître les films de l'ensemble d'entraînement. Même en rajoutant la *data augmentation* et en entraînant suffisamment la tête de classification, nous avons dû faire face à ce problème de sur-apprentissage trop important. L'exemple ci-dessus est le pire que nous ayons rencontré, mais nous avons quand même dû essayer différentes valeurs de taux d'apprentissage et différents protocoles pour savoir comment le faire varier et l'adapter au problème.

Par ailleurs, alors que nous avions au départ essayé de dégeler les couches convolutionnelles une par une, dégeler le ResNet tout entier d'un seul coup s'est avéré être une meilleure solution qui a permis, en utilisant un taux d'apprentissage très faible, à la fois d'avoir une meilleure *accuracy* sur l'ensemble de validation et un sur-apprentissage contrôlé.

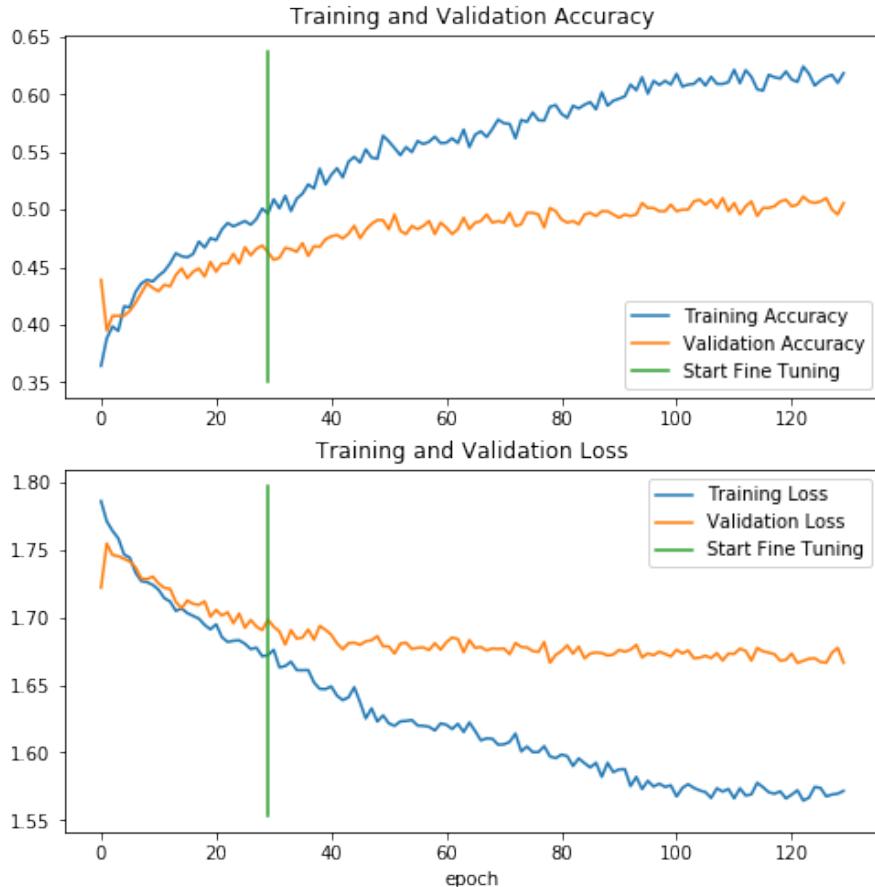


FIGURE 12 – Entraînement de notre meilleur réseau

Cet entraînement s'est fait en utilisant les taux d'apprentissage ci-dessous :

Époques	$1 \rightarrow 15$	$15 \rightarrow 30$	$31 \rightarrow 60$	$61 \rightarrow 100$	$101 \rightarrow 130$
Taux d'apprentissage	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}

Nous obtenons avec ce réseau, une *accuracy* de **0.450** sur l'ensemble de test ce qui est le meilleur résultat que nous ayons obtenu parmi toutes les méthodes testées. Si on regarde plus en détails, on voit que les résultats varient beaucoup suivant les genres alors même qu'ils sont en proportions égales dans l'ensemble d'entraînement et dans l'ensemble de test.

On peut noter directement que le réseau a de très bons résultats sur les films d'animation : 80% d'entre-eux sont correctement prédits, et peu de films d'autre genre sont prédits comme étant des films d'animation. Le réseau a également de bons résultats sur les films d'action et les documentaires, tandis que les comédies dramatiques et les drames sont les genres les plus difficiles à bien prédire.

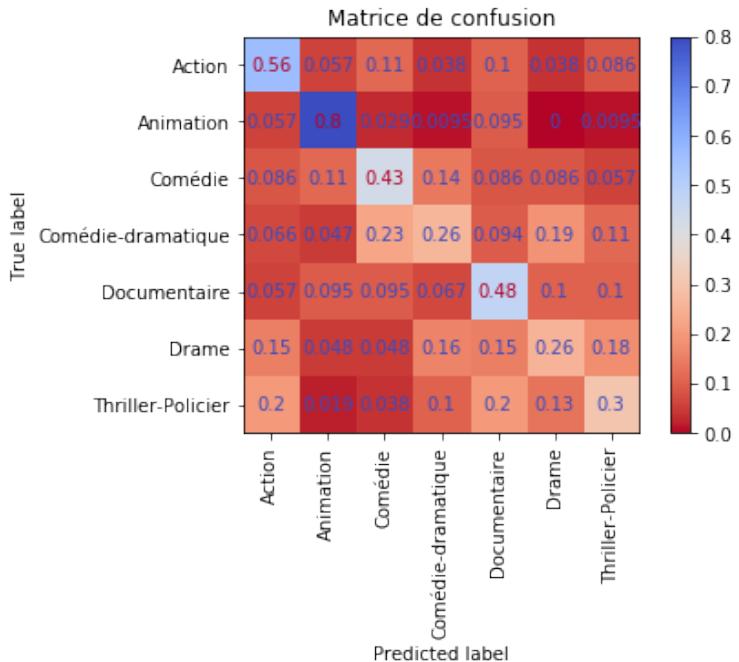


FIGURE 13 – Matrice de confusion pour le réseau final

Tout ces résultats sont cohérents avec ce qu'on sait des genres d'un film : de manière générale, il nous est très facile de reconnaître un film d'animation depuis son poster, et il nous est bien plus dur de reconnaître une comédie dramatique. Les prédictions des comédies dramatiques sont également distribuées en majorité entre Comédie dramatique, Comédie et Drame, ce qui est ce qu'on aurait pu espérer *a priori*.

On peut voir sur l'exemple ci-dessous, des posters de films d'action de l'ensemble de test qui ont été correctement assigné avec une confiance en la prédiction de 100%.



FIGURE 14 – Films d'action prédits avec 100% de confiance

En regardant ces résultats, on peut essayer de déduire ce que le réseau comprend d'un film d'action. Ces posters ont des compositions assez proches. De même, si on regarde les posters équivalents pour les films d'animation, les résultats sont aussi très parlant.



FIGURE 15 – Films d'animation prédits avec 100% de confiance

On peut également chercher quels posters de film d'animation ont été le moins bien prédits. On peut ainsi voir que certains ont été prédits très fortement comme des films d'action et la prédiction est compréhensible : ces posters reprennent les codes et la composition des posters de films d'action. De manière générale, en regardant les résultats, on peut comprendre une proportion importante des erreurs parce que telle affiche "fait documentaire" ou "fait film d'action", etc.

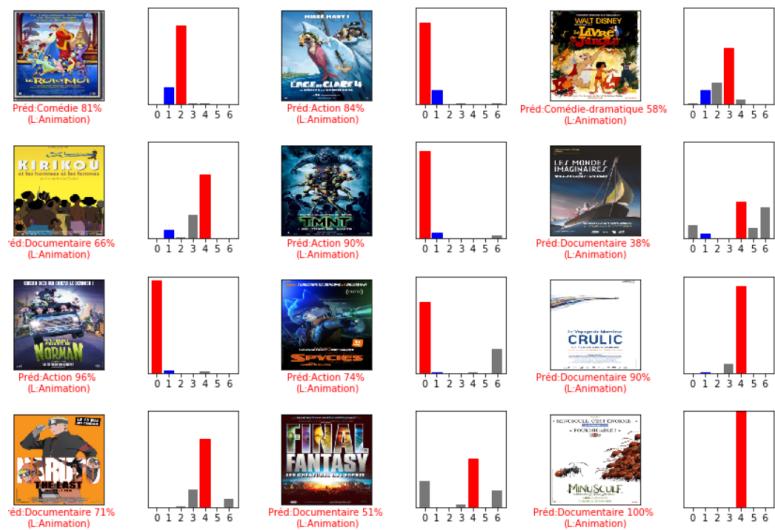


FIGURE 16 – Films d'animation les moins bien prédits

On comprend facilement pourquoi l'*accuracy* sur l'ensemble de test n'est pas plus élevée. La tâche de classer les films suivant leur genre d'après leur poster n'est pas simple :

le genre d'un film n'est pas un attribut strict et les frontières entre les genres sont floues. Même les mauvaises prédictions du réseau sont révélatrices ; et ce réseau serait utile dans un contexte où un studio devrait choisir entre plusieurs affiches pour un film et chercherait à prendre celle qui transmet au mieux le genre du film.

4.3 Arbre de décision

Au cours du projet nous avions essayé décidé d'appliquer une méthode de type *Random Forest* aux *features* extraits par le réseau ResNet. Un intérêt de cette méthode est que les arbres de décisions sont souvent interprétables, comme nous le verrons un peu plus tard dans ce paragraphe. Les résultats obtenus dans un premier temps n'étaient pas très bons, mais après avoir fait le *fine-tuning* du ResNet dans la partie précédente, nous avons voulu réessayer avec ce réseau dont les poids n'étaient plus ceux appris sur ImageNet.

Nous avons implémenté la *random forest* et optimisé ses différents hyperparamètres, en fixant le nombre d'arbres à 200. Nous avons obtenu le meilleur résultat pour des arbres de profondeur 3 et entraînés sur un sixième de l'ensemble d'entraînement. On obtient une précision de **0.420** sur l'ensemble de test, ce qui est moins bien qu'en gardant la tête de classification à la place de la *random forest* mais ce qui ouvre la voie à une interprétation future des *features* du ResNet.

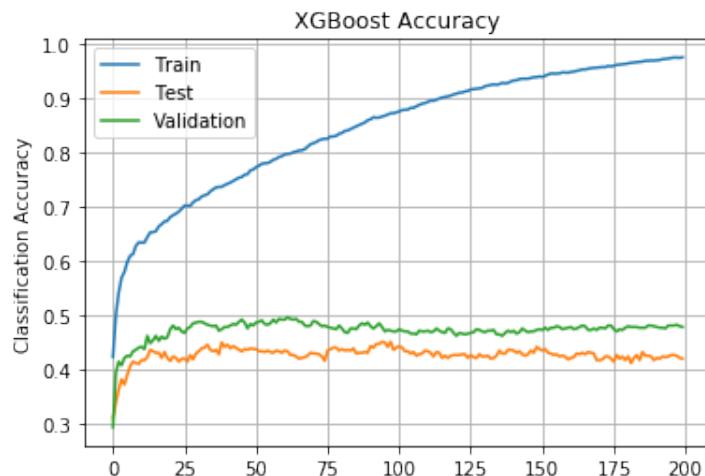


FIGURE 17 – Courbes d'entraînement de notre *random forest*

La matrice de confusion ressemble beaucoup à la matrice précédente, les résultats étant globalement similaires. Les résultats sont très proches pour chaque genre, et on ne distingue pas de phénomène particulier par rapport à précédemment.

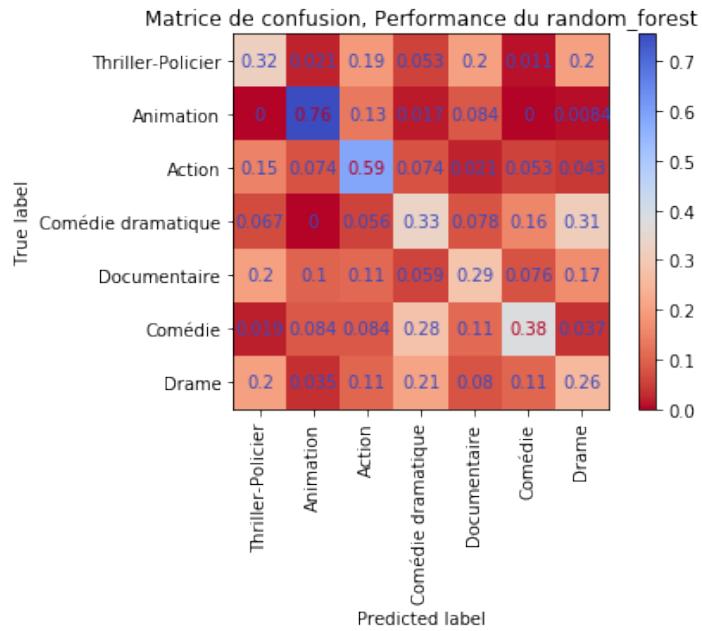


FIGURE 18 – Matrice de confusion de notre *random forest*

Un intérêt des algorithmes de type *random forest* est que l'on peut isoler les *features* les plus discriminantes qu'ils utilisent. La somme des valeurs associées à l'ensemble des *features* vaut 1. On voit que l'intérêt des *features* est plutôt bien réparti. Le plus grand intérêt pour une *feature* vaut à peine plus de 1%, ce qui reste très peu pour l'étudier séparément.

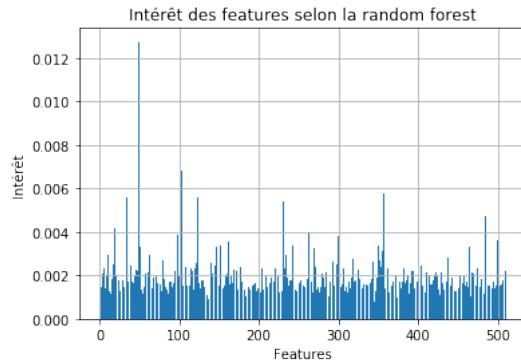


FIGURE 19 – Importance de chacune d'une feature selon notre *random forest*.

Dans notre cas, une étude d'interprétabilité serait intéressante, mais complexe. En effet, rappelons que les *features* sont ici des sorties de réseau de neurones, qui sont peu explicites. Si on trouve qu'une *feature* particulier est important pour la prédiction, il faut ensuite faire une étude du réseau neuronal pour savoir quelle caractéristique du poster met en avant cette *feature*.

Dans le cadre de notre projet, nous n'avons pas eu suffisamment de temps pour nous plonger dans cette étude. Mais c'est une piste intéressante, dans laquelle notre cliente a l'intention de se plonger.

Conclusion

		<i>Test accuracy</i>	Commentaires
Méthode directe	k-NN	16%	Features trop simples
	CNN	20%	Pas adapté à notre base de donnée
<i>Transfer learning</i>	ResNet + k-NN	32%	Contextualisation de la prédiction Aide pour le choix d'une affiche
	ResNet + Random Forest	42%	Perspective d'interprétabilité
	ResNet + Tête de classification	45%	Meilleur modèle Analyses globales sur les genres

Dans un souci de rigueur scientifique, nous avons développé plusieurs méthodes. Pour chaque méthode développée, nous avons choisi des outils pertinents pour mesurer leur qualité, et interpréter leurs résultats. Pour les méthodes qui n'ont pas fonctionné, nous avons compris ce qui les empêchait de fonctionner sur notre problème. Et nous avons également étudié les limites des méthodes peformantes.

Nous avons implémenté des méthodes directes. Elles nous ont permis de prouver que des features bas niveaux n'étaient pas suffisants pour traiter notre problème spécifique, sur la base de donnée que nous avions à disposition.

Réponse aux besoins métiers du client

Nous avons construit, en utilisant du *transfer learning*, un algorithme qui reconnaît le genre principal d'un film avec 45% de fiabilité. A titre de comparaison, l'article [3], qui fait de la prédiction mono-label de genre de poster parmi 4 genres, annonce une fiabilité de 38%. Nous pouvons donc raisonnablement être satisfaits de notre performance. Cet algorithme a tendance à confondre certains genres, comme "Comédie" et "Comédie Dramatique". En réalité, même pour un humain, les affiches associées à ces deux genres ne sont pas toujours dissociables.

En complément du cahier des charges initial, nous livrons également à notre client des outils de visualisation, qui serviront d'aide à la décision.

L'affiche d'un film est en effet un outil marketing crucial. Il faut qu'en un coup d'oeil sur l'affiche, le public sache s'il a envie d'aller voir ce film ou non. Une technique très utilisée dans le milieu est d'utiliser des "codes graphiques" du genre du film. C'est pourquoi beaucoup de comédies françaises sont bleues : cela permet de rapidement identifier le film associé à l'affiche comme une comédie familiale. Ainsi, lorsque le service marketing de Warner Bros France doit créer une affiche pour un film, il est important qu'elle fasse penser à des affiches de films dont le contenu est proche. Il est donc pertinent, pour une affiche candidate, de savoir à quelles affiches déjà existantes elle fait penser. Jusqu'ici, pour répondre à cette question, les créateurs de posters font cette recherche "à la main", en utilisant des catalogues d'affiches.

L'algorithme ResNet+kNN permet d'automatiser cette recherche. Il suffit en effet de donner un poster candidat en argument à l'algorithme, et il retourne les posters de la base de donnée qui en sont le plus proches. Cet algorithme représente donc un gain de temps important pour le service marketing, et permet éventuellement de donner des informations complémentaire, pour aider à la prise de décision. Il pourra servir au service marketing de Warner Bros, pour choisir une affiche, en sachant à quelles autres affiches elle fera penser.

Pistes d'amélioration

Nos résultats sont encourageants, et notre cliente compte poursuivre ce projet. Voici quelques pistes d'amélioration et d'approfondissement, que nous n'avons pas eu le temps de traiter.

Notre travail nous a permis de mettre en exergue les limites de la base de données avec laquelle nous avons travaillé. Il pourrait être pertinent de la compléter avec d'autres bases de données disponibles sur internet. Il faudra néanmoins prendre garde à ne pas ajouter à la base de donnée des posters qui suivent des conventions différentes des conventions françaises.

Le projet de classification de posters est à l'origine un problème *multi-label*. Dans ce rapport, nous avons développé une approche mono-label, et justifié ce choix. Nos algorithmes ont été implémentés de manière pouvoir s'adapter au problème *multi-label*. Vu les bons résultats que nous avons obtenus en mono-label, il semble maintenant réaliste de passer au *multi-label*. La méthode de *transfer learning* qui utilise les forêts aléatoires est une autre piste d'amélioration. Une étude en détail des arbres aléatoires générés pourrait aboutir à une réelle interprétabilité des résultats.

Ce projet de département nous a permis de nous confronter à un problème ludique et riche de *machine learning*. Nous avons pu confronter le contenu de nos cours théoriques à la réalité du terrain. Nous livrons à Warner des algorithmes exploitables, et avons beaucoup appris en les construisant.

Références

- [1] Wei-Ta Chu and Hung-Jui Guo. Movie Genre Classification based on Poster Images with Deep Neural Networks. In *Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes - MUSA2 '17*, pages 39–45, Mountain View, California, USA, 2017. ACM Press. Meeting Name : the Workshop Reporter : Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes - MUSA2 '17.
- [2] davideiacobs. davideiacobs/-Movie-Genres-Classification-from-their-Poster-Image-using-CNNs, February 2020. original-date : 2019-02-24T11:40:40Z.
- [3] Necip Gozuacik and C. Okan Sakar. Turkish movie genre classification from poster images using convolutional neural networks. In *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, pages 930–934.

- [4] Dongmei Han, Qigang Liu, and Weiguo Fan. A new image classification method using CNN transfer learning and web data augmentation. 95 :43–56.
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.
- [6] Marina Ivašić-Kos, Miran Pobar, and Ivo Ipsic. Automatic Movie Posters Classification into Genres. *Advances in Intelligent Systems and Computing*, 311 :319–328, January 2015. Reporter : Advances in Intelligent Systems and Computing.
- [7] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once : Unified, Real-Time Object Detection. *arXiv :1506.02640 [cs]*, May 2016. Reporter : arXiv :1506.02640 [cs] arXiv : 1506.02640.