

PFEE Final Report

Majurca

Abstract This is the final report of our progress on the task of image classification for plastic waste sorting imparted to us by the company Majurca. Seen as this is the final report it will cover everything that has been done since the beginning of the project. This will be done in a chronological manner. As for an overview of what performed has been performed since the beginning of the project; For the image classification task we first chose a pretrained CNN model VGG16 for transfer learning. Then as our first data set was a subset of seven categories each containing 100 images we quickly faced issues regarding overfitting. To combat this employed regularization by adding an additional dropout layer to the VGG16 model. This allowed for a significant increase in validation accuracy (further detailed in subsequent sections). At this point we gained access to the complete dataset of approximately 64000 images. As we tested the previous model on the complete dataset we observed a significant disparity in the precision achieved on the initial dataset and the complete one. Upon further inspection of the data we concluded that this was due to the higher prevalence of images taken from different angles and with different luminosity on the complete dataset than in the initial one. After having informed the company of this change, they notified us that the complete dataset contained outdated images, and showed us some of the up to date ones. Upon further discussion with the company we found that, due to lack of labeling, there was no real way to query the dataset for the up to date images. So in order to retrieve only the up to date images from the dataset we instead implemented an image search engine. Finally, after using the correct images for the image classification task we achieved a final accuracy of 92%.

Jose A. Henriquez Roa
Christopher Diamana Lutete

January 26, 2022

Contents

1	Introduction	1
1.1	Majurca	1
1.2	The task	1
2	Project organization	2
2.1	Our interlocutors	2
2.2	Communication tools	3
2.3	Interaction Stages	3
2.4	Individual task repartition	5
3	The assignment	5
3.1	Task assignment	5
3.2	The work	5
3.2.1	Hand made model architecture	5
3.2.2	Transfer Learning	6
3.2.3	Regularization	7
3.2.4	Content-based image retrieval	9
4	Conclusion	11
4.1	Results	11
4.2	Learnings	11
4.3	Possible Improvements	11
5	Annex	13
5.1	Model Architectures	13

1 Introduction

Cette section présentera l'entreprise Majurca ainsi que la tâche qui nous a été confiée par celle-ci.

1.1 Majurca

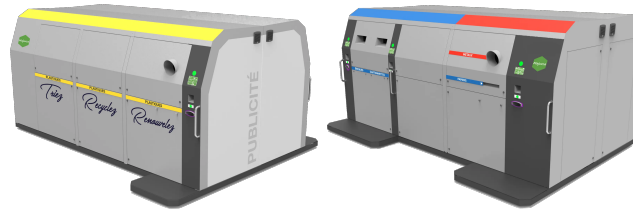


Figure 1: Module de recyclage du plastique (à gauche). Modules de recyclage du métal (à droite)

La société Majurca a été fondée le 8 octobre en 2013 par Stéphanie Loss Letienne, il y a actuellement deux gérants Benjamin Letienne et Valérie Paris Letienne. L'entreprise elle-même est répertoriée dans le secteur d'activités de la fabrication de machines spécialisées. Le principal projet actuellement mené par l'entreprise est celui de la création d'un ensemble de terminaux modulaires de classification et de broyage automatique des déchets plastiques pour en faciliter le recyclage. Les modules de recyclage du plastique et du métal sont présentés dans la figure 1.

Les deux machines prennent les déchets plastiques ou métalliques à recycler par l'ouverture circulaire en haut à droite. Selon le type de plastique ou de métal, elles broient d'abord les déchets et les trient dans des conteneurs respectifs. Une fois pleins, ces conteneurs seront ensuite sortis et transportés vers des centres de recyclage. Les deux principaux points mis en avant par la société Majurca concernant cette nouvelle méthode de recyclage sont les suivants :

1. Le fait que les modules automatisent le tri des déchets au niveau le plus granulaire effectivement requis avant le broyage du plastique (par exemple selon le type de plastique ou de métal).
2. Et le fait qu'en broyant les déchets sur place, on peut exporter directement les déchets broyés vers des centres de recyclage.

Les deux machines semblent encore en être aux premiers stades de développement, celle destinée au recyclage du plastique étant plus avancée. Sur les deux, celle à laquelle notre groupe a travaillé avec était exclusivement celle du recyclage du plastique. Notre tâche était liée à l'étape de reconnaissance du type de plastique au tout début de la pipeline.

1.2 The task



Figure 2: Top angle (Left). Side angle (Right)

Le mécanisme qui est actuellement envisagé pour la tâche de classification des déchets est la classification d'images par des modèles d'apprentissage profond. Notre tâche consistait à mettre en œuvre un tel modèle.

Les images sont prises selon le processus suivant : les déchets plastiques sont d'abord déposés dans une chambre située juste après l'ouverture circulaire en haut à droite des machines. Deux images sont ensuite prises pour chaque déchets déposé sous différents angles. Une vue de dessus et une vue de côté. Vous trouverez deux exemples sur la figure 2. Notre tâche consiste alors à entraîner des modèles de classification d'images pour détecter le type de plastique des objets insérés dans la chambre. Pour cela, nous pouvons soit former un seul modèle pour détecter le type de plastique malgré l'angle, soit former deux modèles distincts. Un pour chaque angle d'image. Ce point est discuté plus en détail dans les sections suivantes.

C'est l'étendue de la tâche qui nous a été confiée par Majurca. Il s'agissait d'une simple tâche de classification d'images. La partie classification elle-même était assez simple. Là où nous avons rencontré le plus de complications en travaillant sur le projet, c'est dans la communication avec l'entreprise elle-même. Ces problèmes seront détaillés dans les sections suivantes. Cependant, nous notons que tout a fini par s'arranger.

2 Project organization

Cette section présentera tout ce qui concerne notre communication avec la société Majurca ainsi que la répartition des tâches de chacun des membres de notre groupe. Tout d'abord, nous présenterons le représentant de Majurca qui était en charge de la communication avec nous. Nous présenterons ensuite les outils de communication qui ont été utilisés lors de nos interactions. Ensuite, nous détaillerons les étapes d'interaction que nous avons eues avec l'entreprise. Et enfin, nous présenterons chacun des membres de notre groupe ainsi que les tâches dont nous étions individuellement chargés.

2.1 Our interlocutors

Tout au long du projet, nous avons eu plusieurs interlocuteurs du côté de l'EPITA, et un seul du côté de l'entreprise.

La représentante de la société qui a supervisé notre travail tout au long du projet était Stéphanie Loss Letienne. Comme mentionné précédemment, elle est la fondatrice de la

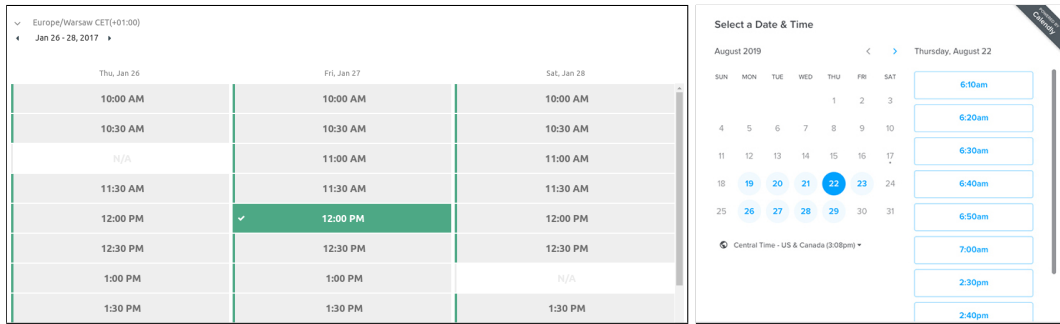


Figure 3: Appoint.ly (gauche), Calendly (droite)

société Majorca. Elle s'est chargée à la fois de l'attribution des tâches et de la supervision de notre progression.

Du côté d'epita, nous avons eu trois interlocuteurs, chronologiquement ce sont :

1. Pierre-Julien Grizel a été le premier à diriger notre interaction avec l'entreprise. Il était présent lors des premières réunions avec celle-ci. Au cours de ces réunions, il nous a aidés, nous et l'entreprise, à régler quelques détails techniques, notamment en ce qui concerne l'accès au jeu de données d'images. Au cours de ces réunions, il a également aidé à répondre à certaines des questions de Stéphanie concernant les modèles de Deep Learning pour la classification d'images, comme la consommation d'énergie et le temps d'inférence.
2. Cédric Joly était notre deuxième interlocuteur du côté de l'EPITA. Au moment du changement, nous étions déjà en charge de la plupart des interactions avec l'entreprise depuis un certain temps. Pierre-Julien Grizel continuait cependant à nous aider pour tout ce qui concernait le jeu de données d'images. Principalement en nous donnant des indications sur la manière d'accéder à celle-ci par e-mail. Juste après le changement, nous avons été informés que nous devions faire des rapports mensuels lors de réunions Teams sur tout ce qui s'était passé au cours du mois avec l'entreprise. Après avoir informé Cédric Joly de tout ce qui s'était passé avec l'entreprise depuis le début du projet, il nous donnait des conseils supplémentaires sur de nouvelles méthodes pour améliorer notre précision dans la tâche de classification d'images. Ces réunions ont duré environ trois mois.
3. Nicolas Boutry a été notre dernier interlocuteur. Le changement d'interlocuteur a eu lieu juste avant que la dernière réunion avec Cédric Joly ne soit programmée (qui devait également être la toute dernière avant la présentation finale) et la plupart des interactions avec lui se sont donc faites par e-mail.

2.2 Communication tools

Trois outils ont été utilisés pour la communication entre l'entreprise et les interlocuteurs de l'EPITA.

Il s'agit de Teams, d'Emails et du site Whereby.com pour les appels vidéo. Nous décrivons ci-après comment ces outils de communication ont été utilisés et avec qui.

1. *Whereby.com* C'est le premier outil que nous avons utilisé pour communiquer avec Stéphanie Loss Letienne et Pierre-Julien Grizel. C'est ici qu'ont eu lieu les premières rencontres avec Pierre-Julien Grizel ainsi que toutes celles qui ont suivi avec Stéphanie Loss Letienne.

Whereby est un site web d'appel vidéo qui annonce une plus grande facilité d'utilisation par rapport à d'autres applications d'appel vidéo. Avec Whereby, les réunions sont organisées directement sur leur site web. Les clients n'ont qu'à réserver une URL, qui sera ensuite partagée avec tous les membres qui participeront à la réunion.

2. *Email* La plupart des communications concernant les horaires des réunions par vidéo-conférence ainsi que les communications concernant certains des détails techniques liés aux tâches de classification des images (par exemple, les questions relatives aux ensembles de données) ont été effectuées par courriel.
3. *Teams* Cette application a été utilisée exclusivement pour les rapports mensuels avec Cédric Joly.

Deux autres outils ont été utilisés pour coordonner les dates et heures des réunions. Il s'agit de Appoint.ly et Calendly.

1. *Appoint.ly* Cette application a été utilisée pour coordonner les premières rencontres avec Pierre-Julien Grizel et Stéphanie Loss Letienne. Une capture d'écran de l'interface de cette application se trouve sur la figure 3 à gauche.
2. *Calendly* Celle-ci a été utilisée pour coordonner les réunions de Teams avec Cédric Joly. Une capture d'écran de l'interface de cette application se trouve sur la figure 3 à droite

2.3 Interaction Stages

Pour notre interaction avec la société Majurca, nous comptons quatre étapes principales. Celles-ci ont été séparées par les moments où nous avons eu accès à différentes parties de l'ensemble de données d'images.

1. *Définition de la tâche et Ensemble de données initial* Cette étape commence avec la première réunion. Au cours de celle-ci, nous avons reçu plus de détails sur ce que serait la mission. Par exemple, on nous a montré pour la première fois des exemples d'images que nous utiliserions pour la tâche de classification tout au long du projet.

Nous avons ensuite été informés de l'endroit où le modèle de classification serait utilisé, c'est-à-dire à l'intérieur des modules de recyclage. On nous a ensuite montré comment les images étaient prises à l'intérieur de la machine et à quel type de variations nous pouvions nous attendre d'une image à l'autre, afin que nous puissions en tenir compte pendant l'entraînement.

Après cette première réunion, nous avons reçu un petit sous-ensemble de l'ensemble de données composé de 700 images réparties en sept catégories de 100 images chacune.

C'est ainsi que s'est terminé la première rencontre, et donc la première étape.

2. *Rapport d'avancement de l'ensemble de données initial* Après avoir obtenu l'accès à l'ensemble de données initial, nous avons également eu le temps de tester quelques modèles initiaux de classification. Cette étape s'est déroulée au cours des trois réunions suivantes, soit environ un mois après la réunion initiale. Au cours de chacune de ces réunions, nous avons présenté chacun des modèles de classification que nous avons testés. Nous avons également présenté certaines des méthodes de régularisation que nous avons utilisées pour traiter le petit ensemble de données. À la fin de chaque réunion, nous indiquions lequel des modèles que nous avons testés avait donné les meilleurs résultats pour le projet.
3. *Ensemble de données complet* Après avoir produit quelques résultats initiaux, nous avons eu accès à l'ensemble de données complet. Celui-ci était composé d'un nombre beaucoup plus élevé d'images ainsi que d'un nombre beaucoup plus élevé de catégories. Il y avait cependant quelques problèmes dans la qualité des nouvelles images. Nous y reviendrons à ce sujet dans les sections suivantes. Après avoir obtenu l'accès au jeu de données complet, nous sommes entrés dans la phase finale suivante.
4. *Rapport d'avancement continu* Après avoir obtenu l'accès à l'ensemble des données complet, il ne restait plus qu'à continuer à entraîner et à modifier les modèles de manière itérative afin d'obtenir la plus grande précision possible. Cette étape s'est poursuivie jusqu'à la toute dernière réunion. Au cours de chaque réunion, nous avons présenté les améliorations de la précision que nous avons obtenues au cours des dernières versions. Nous avons également présenté les méthodes que nous avons employées pour améliorer les derniers modèles (par exemple, augmentation des données, régularisation, modifications de l'architecture des ANN, etc.)

2.4 Individual task repartition

Notre groupe a commencé avec trois personnes : Felix Auneau, Christopher Diamana Lutete et José A. Henriquez Roa. Cependant, peu après la deuxième rencontre avec Majurca Felix Auneau semble avoir quitté l'école. Ainsi, cette section ne présentera que la répartition du travail entre Christopher Diamana Lutete et Jose A. Henriquez Roa. Les sections suivantes seront présentées par leur individu respectif :

- Jose A. Henriquez Roa : Au cours du premier semestre du projet, c'est Felix Auneau et moi qui avons contribué à la formation et au test des modèles de classification. Felix Auneau a fait la plupart du travail pour la première réunion et j'ai pris en charge le travail des réunions suivantes après son départ. Ensuite, pour mieux équilibrer la répartition du travail entre les deux membres restants pour le second semestre, une décision a été prise d'alterner le travail qui devait être fait avant chaque réunion entre Christopher Diamana Lutete et moi.
- Christopher Diamana Lutete : Au cours du second semestre du projet, j'ai essayé d'améliorer les modèles de classification. J'ai également nettoyé le jeu de données en détectant les images mal étiquetées.

3 The assignment

Cette section présente une description détaillée de tout ce qui a été fait pour la tâche de classification d'images qui nous a été confiée par Majurca. Elle présente d'abord la tâche

telle que définie par la société. Puis la description détaillée de notre travail. La section suivante présentera une conclusion à ce rapport en présentant d’abord les résultats finaux ainsi que les améliorations possibles qui pourraient encore être mises en œuvre avec un peu plus de temps et de ressources.

3.1 Task assignment

Lors de la première réunion, la tâche qui nous a été confiée était assez simple. Nous devions mettre en œuvre un modèle de classification d’images qui, lorsqu’on lui donnait une photo d’un objet en plastique, devait déterminer le type de plastique de cet objet. Pour les différentes catégories de types de plastique, nous en avons reçu sept au total. Il n’y a pas eu de problèmes ou de malentendus quant à la tâche à accomplir.

3.2 The work

Cette section présente chronologiquement tout ce qui a été fait pendant le projet. Chaque nouvelle sous-section présentera les méthodes qui ont été utilisées pour améliorer les problèmes rencontrés dans la précédente.

3.2.1 Hand made model architecture

Après avoir reçu le jeu de données initial de 700 images, nous avons d’abord décidé d’essayer de voir quel genre de résultats nous obtiendrions en utilisant un modèle fait à la main. Dans des scénarios plus limités dans le temps, il aurait été plus logique de passer directement à l’apprentissage par transfert. Cependant, comme nous savions que le projet durerait environ deux semestres, nous avons voulu être aussi exhaustifs que possible dès le départ.

Le modèle lui-même était composé de deux couches de convolution et d’une couche FC (fully connected). Au total, il y avait 1,699,399 paramètres entraînaables. Après avoir testé un nombre considérable de variantes, nous avons trouvé que celle-ci était la plus performante pour l’ensemble d’entraînement. Nous avons essayé de garder la quantité de paramètres entraînaables aussi petite que possible et l’architecture aussi peu profonde que possible afin de combattre l’overfitting autant que possible. Une description complète du modèle peut être trouvée sur la figure 11.

Cependant, étant donné que l’ensemble de données initial était assez petit et par le fait que trouver manuellement une architecture qui ne overfit pas est assez rare et prend du temps. Nous n’avons pas fini par trouver des résultats acceptables sur le jeu d’images de test. En moyenne, nous avons trouvé 8% de précision sur celui-ci.

Nous avons donc décidé d’abandonner l’idée de créer un modèle manuellement dans l’espoir d’obtenir de meilleurs résultats grâce à l’apprentissage par transfert.

3.2.2 Transfer Learning

L’apprentissage par transfert consiste à utiliser d’autres réseaux neuronaux artificiels (ANN) qui ont été formés pour des tâches distinctes mais similaires. Et on modifie progressivement les paramètres appris en réentraînant le réseau sur un nouvel ensemble de données afin de le rendre utile pour une tâche différente de celle d’origine. Les réseaux les plus populaires

qui sont choisis pour l'apprentissage par transfert sont ceux qui sont connus pour obtenir de bon résultats sur des tâches similaires à celle qui est abordée et qui sont capables de généraliser ce qu'ils ont appris à d'autres tâches jamais vues.

Dans le cas de la de computer vision, les réseaux neuronaux pré-entraînés sont généralement évalués en fonction de leurs performances sur le jeu de données ImageNet. Ceci est dû au fait que le jeu de données ImageNet est composé de plus de 18 million d'images avec plus de 20 millier de catégories d'images qui ont très peu en commun les unes avec les autres, comme "ballons", "fraises", "chien" et plus encore. Si un réseau obtient de bons résultats dans une telle tâche, il est presque certain qu'il sera capable de généraliser à d'autres tâches.

Pour notre tâche, nous avons testé tous les réseaux les plus performants sur la tâche ImageNet, une liste de tous les modèles que nous avons testés peut être trouvée sur <https://keras.io/api/applications/>, ici vous trouverez des informations sur l'achitecture de chaque modèle ainsi que leur précision respective sur ImageNet.

Parmi tous les modèles que nous avons testés, ceux qui nous ont semblé les plus prometteurs pour la tâche actuelle sont VGG16, InceptionV3, ResNet50, EfficientNetB0 et Xception. Et parmi ceux-ci, celui que nous avons fini par choisir était VGG16. Bien qu'il ne soit pas le plus performant sur ImageNet, le VGG16 est relativement petit, ce qui signifie qu'il est le plus facile à entraîner parmi ceux qui ont été considérés précédemment. Le VGG16 est également très connu pour sa capacité à généraliser très bien sur un nombre relativement faible d'époques par rapport aux autres modèles. C'est également l'un des facteurs qui nous a conduit à faire ce choix. Il est cependant important de noter que certains des autres réseaux sont encore susceptibles de surpasser le VGG16 pour la tâche actuelle si on leur donne un nombre d'époques d'entraînement beaucoup plus élevé. Nous avons choisi d'utiliser le VGG16 principalement en raison du manque de ressources informatiques.

En dehors des couches de Pooling, VGG16 est composé de 16 couches, 13 couches convolutionnelles et 3 couches entièrement connectées (FC). Avec un total de 138,357,544 paramètres entraînaibles. Les détails complets de l'architecture peuvent être trouvés sur la figure 12. Sans modification supplémentaire du modèle, nous avons obtenu une précision de 75% sur l'ensemble de validation, ce qui représente une amélioration significative par rapport à la dernière version. La figure suivante montre la courbe d'apprentissage de ce modèle particulier au fil des 300 époques :

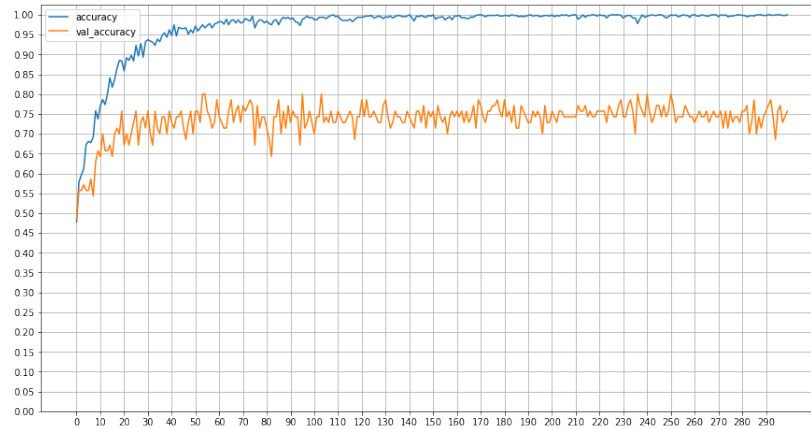


Figure 4: Courbe d'apprentissage du VGG16

Comme on peut le voir dans la figure 4, il y a toujours une grande séparation entre la précision de validation et la précision de test. Cela signifie que le modèle est toujours surajusté (overfit) aux données d'apprentissage. Pour remédier cela, nous avons appliqué de la régularisation, comme le présente la section suivante.

3.2.3 Regularization

L'application de régularisation à un modèle ANN consiste à contraindre les poids que le réseau tente d'apprendre pendant l'entraînement. Cela peut se faire par le biais d'un certain nombre de méthodes différentes. Les méthodes les plus couramment utilisées pour les réseaux neuronaux profonds sont la régularisation L1, la régularisation L2 et le Dropout. Selon la méthode utilisée, le fait de contraindre les poids peut avoir des effets différents.

Certaines régularisations aident à résoudre le problème de la disparition et de l'explosion des gradients, fréquemment rencontré dans les réseaux neuronaux. Le problème de la disparition et explosion de gradient fait référence au fait que, selon les données d'entrée, les gradients calculés utilisés pour mettre à jour les poids du réseau peuvent soit être évanescents, ce qui signifie qu'ils n'auront pratiquement aucun impact sur la mise à jour du réseau, et que c'est comme si les données n'avaient jamais été utilisées. Ou bien les gradients peuvent être trop grands (exploding), ce qui signifie que les poids auront tendance à écraser complètement toute valeur antérieure déjà présente dans le réseau, forçant ainsi le réseau à oublier toute donnée antérieure, ce que l'on appelle également interférence catastrophique ou oubli catastrophique. Les méthodes connues pour résoudre ce problème particulier sont les régularisations L1 et L2, ainsi qu'une autre méthode appelée normalisation par batch. Cette dernière n'est cependant pas une simple méthode de régularisation, mais elle aide néanmoins à résoudre ce problème.

Avec le dernier modèle, il n'y avait pas de problème de disparition ou d'explosion de gradient. Il y avait cependant un surajustement (overfit) significatif. Un modèle qui overfit est un modèle qui donne de bons résultats sur l'ensemble d'apprentissage mais des résultats relativement mauvais sur l'ensemble de validation. En d'autres termes, il mémorise les données d'apprentissage et, en faisant cela, il n'est pas en mesure de généraliser correctement à des données qu'il n'a pas vues. La régularisation peut également aider à résoudre

ce problème. L'une des méthodes de régularisation les plus efficaces contre l'overfitting est le dropout. Cette méthode tire son nom du fait que tout réseau qui la met en œuvre verra certains de ses nœuds désactivés ou éliminés pendant l'entraînement selon une probabilité prédéfinie. C'est un peu difficile d'expliquer la manière dont le dropout contribue à réduire l'overfitting. Cependant, on peut interpréter que le dropout entrave la tendance d'un réseau à mémoriser les données d'apprentissage en forçant ces données à être apprises par de multiples nœuds aléatoires répartis dans tout le réseau. Cela empêche à son tour la possibilité que plusieurs nœuds voisins soient codépendants et mémorisent tous ensemble la même portion des données d'entrée.

Il convient toutefois de préciser que les méthodes de régularisation augmentent souvent le temps nécessaire à un modèle de machine learning pour converger vers une bonne solution. C'est particulièrement vrai dans le cas de dropout. Cela se traduit par un besoin accru de ressources informatiques ou de temps pour entraîner les modèles.

Pour revenir au sujet de notre modèle. Après avoir testé de nombreuses configurations de couches de dropout sur l'architecture initiale. Nous avons trouvé les meilleurs résultats en remplaçant simplement l'une des couches FC par une couche de dropout sur le modèle VGG16. D'autres modifications ont été effectuées aux modèles en ce qui concerne les couches entièrement connectées. Nous avons constaté que pour compenser l'augmentation du temps d'apprentissage induit par la couche de dropout, il serait préférable de diminuer le nombre de nœuds des couches entièrement connectées du modèle VGG16. Après ces modifications du modèle, nous avons diminué le nombre de paramètres entraînables de 138,357,544 à 14,847,815, l'architecture complète se trouve sur la figure 13. Et la précision est passée de 75% à 88%. Le graphique suivant montre une comparaison en entre les courbes d'apprentissage du modèle initial et du nouveau modèle :

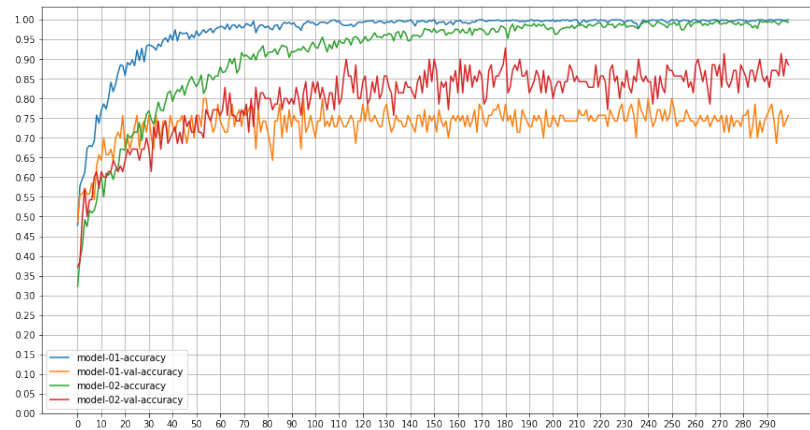


Figure 5: Comparaison de VGG16 et VGG16 avec dropout. La précision de formation et la précision de validation du modèle initial sont respectivement les courbes bleue et orange. La précision de formation et la précision de validation du nouveau modèle sont respectivement les courbes verte et rouge.

Comme le montre le graphique ci-dessus, la précision de validation du nouveau modèle dépasse rapidement celle du modèle précédent pendant l'apprentissage. Cependant, même avec la diminution considérable du nombre de paramètres entraînables, le modèle prend

toujours beaucoup plus de temps que le modèle initial pour converger vers une solution. Une fois de plus, cela est dû à la couche de dropout.

3.2.4 Content-based image retrieval

À partir de ce moment, la plupart des améliorations de la précision ont été obtenues en modifiant les données donné au réseau.

Comme mentionné précédemment, nous avons initialement eu accès à un jeu de données composé d'environ 700 images. Nous avons ensuite eu accès à un jeu de données complet d'environ 64 000 images. Après avoir entraîné le modèle sur le nouveau jeu de données, nous avons remarqué une légère baisse de la précision du modèle par rapport à celle obtenue sur le jeu de données initial. Au départ, nous avions pensé que cela était dû au fait que les échantillons de l'ensemble de données initial étaient très similaires les uns aux autres et qu'il était donc plus facile de généraliser ce qui avait été appris sur l'ensemble d'entraînement à l'ensemble de validation. Cependant, après avoir examiné les nouvelles données, nous avons remarqué qu'il y avait de légères différences dans les images entre l'ancien et le nouveau jeu de données.

Comme mentionné précédemment, les modules étaient censés prendre des images sous deux angles, l'un latéral et l'autre horizontal. Dans le nouvel ensemble de données, nous avons constaté que ces images présentaient de légères différences par rapport à l'ensemble de données précédent:

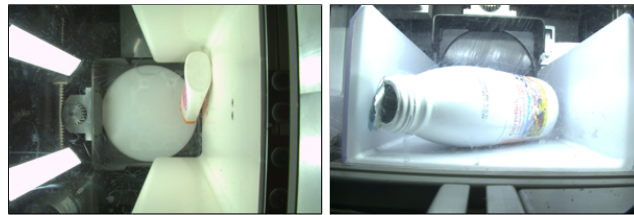


Figure 6: Image taken from the side.

Pour les images prises sur le côté, nous avons remarqué que certaines étaient prises horizontalement et d'autres verticalement. Pour certaines images prises sur le côté, nous avons également remarqué de légères différences dans la teinte des images. Pour les images prises de côté, certaines avaient une teinte plus verdâtre que d'autres. Ceci est légèrement présent dans l'image à gauche de la figure 6.



Figure 7: Image taken from the top.

Pour les images prises depuis le haut, les différences sont plus subtiles. Dans la figure 7,



Figure 8: Recherche d'images basée sur le contenu des angles latéraux



Figure 9: Recherche d'images basée sur le contenu sous un angle supérieur

nous voyons qu'il y a eu quelques légers changements dans le positionnement de la caméra. Nous avons également trouvé quelques différences supplémentaires en ce qui concerne la teinte. En effet, certaines images prises du haut avaient une teinte plus jaune que d'autres.

Après avoir posé quelques questions à ce sujet, on nous a dit que l'ensemble de données contenait des images provenant à la fois d'anciennes machines, qui avaient des angles de caméra différents à l'époque, et de nouvelles machines, avec des angles et un éclairage différents. Cependant, en regardant le fichier SQL que nous avons utilisé pour télécharger l'ensemble des données, nous avons constaté qu'il n'y avait pas de balise indiquant de quelle machine provenait chaque image. Nous n'avions que des exemples de ce à quoi ressemblaient les images les plus récentes. Et comme nous n'avions pas le temps de parcourir l'ensemble des données à la recherche d'images similaires, nous avons décidé de mettre en œuvre une recherche d'images basée sur le contenu pour rechercher les images similaires dans l'ensemble des données.

Pour cela, nous avons utilisé une bibliothèque python appelée DeepImageSearch qui implémente à la fois l'indexation et la logique de recherche lorsqu'on lui donne un jeu de données d'images en entrée. Dans les figures 8 et 9 les deux exemples que nous avons utilisés pour la requête sont les plus à gauche et tous les autres sont les résultats de la recherche.

Le résultat de la recherche était un ensemble d'index d'images pointant vers les images les plus similaires aux images de la requête, toutes triées par similarité. Puisque les images de la requête provenaient de la deuxième machine, il ne restait plus qu'à parcourir les images et à trouver manuellement les points de coupure où les images commençaient à provenir de l'ancienne machine. Après avoir fait cela pour les sept catégories, nous avons obtenu un ensemble de données composé uniquement des images de la deuxième machine.

Après avoir supprimé les anciennes images de l'ensemble de données, nous en avons obtenu un nouveau ensemble de données composé d'environ 53 000 images. En ré-entraînant le modèle sur le nouveau jeu de données, nous avons obtenu une précision d'évaluation de 92%. Avec la courbe d'apprentissage suivante :

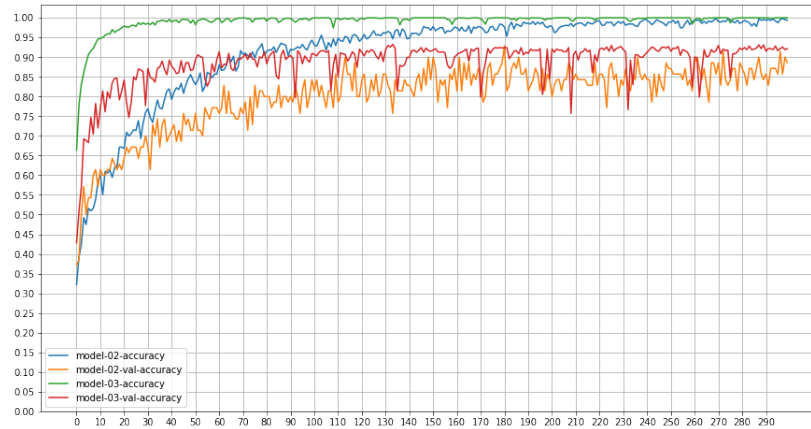


Figure 10: Comparaison du modèle VGG16 avec dropout avec l’ancien et le nouvel ensemble de données. La précision de formation et la précision de validation du modèle formé sur l’ancien jeu de données sont respectivement les courbes bleue et orange. La précision de formation et la précision de validation du modèle formé sur le nouveau jeu de données sont respectivement les courbes verte et rouge.

Comme le montre la figure 10, la suppression des images de l’ancienne machine a considérablement augmenté le temps de convergence. Cela peut être dû au fait que, bien qu’il s’agisse d’images des mêmes objets mais avec des angles légèrement différents, les anciennes images ont dû agir comme du bruit plutôt que de contribuer à la tâche de classification.

4 Conclusion

4.1 Results

En utilisant l’apprentissage par transfert et le Dropout pour la régularisation et en retirant les anciennes images de l’ensemble de données par le biais de Content based image retrieval, nous avons pu obtenir une précision de 92% sur la tâche de classification d’images donnée.

4.2 Learnings

Grâce à ce projet, nous avons pu appliquer tout ce que nous avons appris pendant les cours à une tâche de classification d’images d’une durée de deux semestres. Grâce à celui-ci, nous avons pu utiliser toute la théorie qui nous a été donnée pendant les cours pour nous attaquer à une tâche du monde réel. Par exemple, nous avons pu mettre en pratique les thèmes de la vision par ordinateur et de la content-based image retrieval que nous avons respectivement appris pendant nos cours d’introduction aux réseaux neuronaux et d’apprentissage automatique pour la reconnaissance de pattes.

4.3 Possible Improvements

En plus d’être limités en termes de temps que nous pouvions consacrer à ce projet en raison des autres projets et cours universitaires, nous étions également quelque peu limités en termes de ressources informatiques. Comme nous n’avons accès qu’à nos machines locales pour ce projet, nous étions surtout limités par notre matériel personnel. Cela s’est

traduit par une incapacité à tester certains des modèles les plus avancés et complexes. C'est pourquoi nous avons gardé la VGG16. Nous pensons donc que si nous avions pu avoir accès à des services de cloud, nous aurions pu améliorer nos chances d'obtenir une plus grande précision.

La dernière chose qui aurait pu servir d'amélioration aurait été de poser plus de questions concernant l'ensemble de données dès le début du projet. Ce n'est qu'assez tard au cours du projet que nous avons fini par découvrir que l'ensemble de données contenait une quantité importante d'images vieilles. Si nous l'avions découvert plus tôt, nous aurions pu former les modèles initiaux avec les bonnes données. Ce qui nous aurait permis d'apporter des modifications à l'architecture qui, à leur tour, auraient permis d'obtenir une meilleure précision finale.

5 Annex

5.1 Model Architectures

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 200, 300, 32)	896
max_pooling2d (MaxPooling2D)	(None, 100, 150, 32)	0
conv2d_1 (Conv2D)	(None, 100, 150, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 50, 75, 64)	0
flatten (Flatten)	(None, 240000)	0
dense (Dense)	(None, 7)	1680007
Total params: 1,699,399		
Trainable params: 1,699,399		
Non-trainable params: 0		

Figure 11: Hand made model

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000
=====		
Total params: 138,357,544		
Trainable params: 138,357,544		
Non-trainable params: 0		
=====		

Figure 12: VGG16 model

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, None, None, 3)]	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73856
block2_conv2 (Conv2D)	(None, None, None, 128)	147584
block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295168
block3_conv2 (Conv2D)	(None, None, None, 256)	590080
block3_conv3 (Conv2D)	(None, None, None, 256)	590080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1180160
block4_conv2 (Conv2D)	(None, None, None, 512)	2359808
block4_conv3 (Conv2D)	(None, None, None, 512)	2359808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0
block5_conv1 (Conv2D)	(None, None, None, 512)	2359808
block5_conv2 (Conv2D)	(None, None, None, 512)	2359808
block5_conv3 (Conv2D)	(None, None, None, 512)	2359808
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 256)	131328
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 7)	1799
=====		
Total params: 14,847,815		
Trainable params: 14,847,815		
Non-trainable params: 0		
=====		

Figure 13: VGG16 model with dropout