PROJET CARREFOUR CHALLENGE DE RECONNAISSANCE D'IMAGES

Projet de MAP569

21 mars 2021

Paul MATHIVON, Victor SAILLANT, Geoffrey SAUNOIS
promotion 2018

ÉCOLE POLYTECHNIQUE
IP PARIS

Carrefour

TABLE DES MATIÈRES

INTRODUCTION

Ce travail de reconnaissance d'image est proposé par l'entreprise Carrefour, en collaboration avec le cours de *Machine Learning II*, identifié *MAP 569*.

Le but de ce projet était d'arriver à reconnaître, à partir d'une photo d'un produit prise par un consommateur, le produit du catalogue Carrefour associé. Pour cela, la base de données de tous les produits Carrefour est donnée, avec des photos de différents angles de chaque produit.

Le principal défi de ce challenge était la taille très conséquente de la base de données, qui contenait presque 100G d'images pour un total de 80000 références. Ceci impose de mettre en place des architectures de données pertinentes, capable de supporter un nombre de données important et d'évoluer à grande échelle.

Une des application les plus prometteuse de se projet serait dans la création d'une application pour frigos connectés, où une caméra à l'intérieur du frigo serait capable de reconnaître les aliments encore présents dans le frigo, et ainsi de proposer au consommateur une liste de courses personnalisée.

1
PRÉSENTATION DU PROBLÈME

1.1 Présentation

Concrètement, l'objectif dans ce challenge est de créer un modèle pouvant reconnaître, à partir d'une photo prise par un consommateur, le produit du catalogue Carrefour associé, comme représenté en figure 1.

Figure 1 – Exemple de reconnaissance d'image

Cependant, les ressources à disposition n'étaient pas suffisante pour l'entraînement d'une architecture sur une quantité de données aussi imposante que celle communiquée. En particulier il est impossible de relier directement un produit à son identifiant avec le dataset fourni : il ne contient que 4 à 6 images par produit et rendu l'utilisation du machine learning corrompue.

À la place, il est décidé de regrouper les produits par *catégories*, ce qui permettait ainsi d'augmenter grandement le nombre de données par classes, et ainsi d'entraîner de manière plus viable un modèle.

1.2 Cadre retenu

1.2.1 • Calcul des labels

Afin de traiter ce problème, Carrefour donne également accès aux méta-données associées à chaque produit du catalogue. En plus des code-barres (qui faisaient office d'identifiant pour les produits, et permettant de relier ces informations aux images correspondantes), ces meta-données informent également des catégories aux quelles appartenaient chaque produit.

Les catégories sont **organisées sous forme d'arbre** : une catégorie principale contient plusieurs catégories secondaires, qui elles-mêmes contiennent plusieurs catégories tertiaires... En itérant sur toutes les données, nous avons obtenus un arbre similaire à celui représenté en figure 2. Ainsi, plutôt que d'essayer de relier une photo d'un produit au produit exact correspondant dans la base de données, ce qui posait problème comme vu précédemment, le choix est fait de seulement essayer d'identifier la catégorie (la plus précise) à laquelle ce produit appartient.

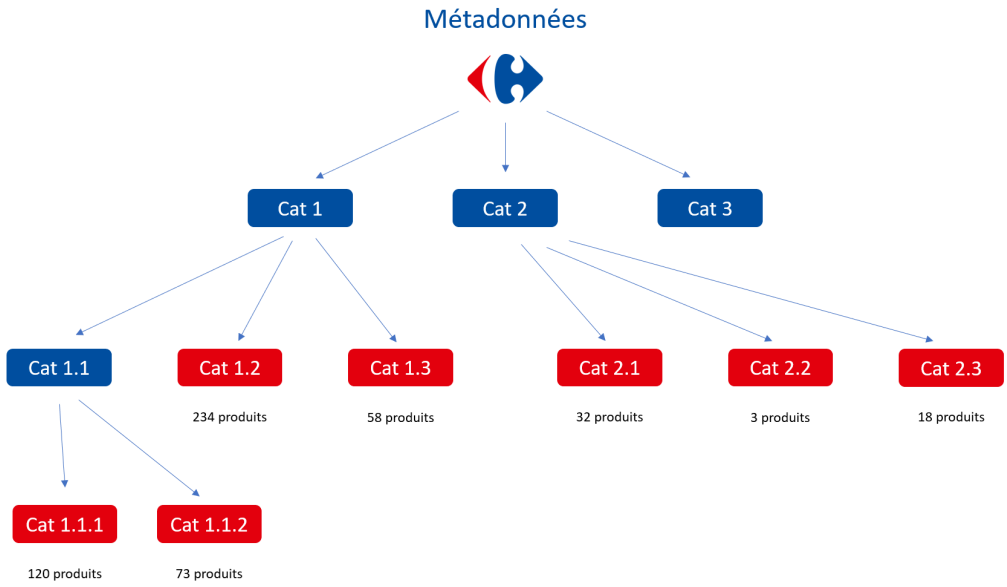

Figure 2 – Labels avant pré-processing

Cependant, cette approche avait le défaut que parmi les catégories "feuilles" de l'arbre (représentées en rouge sur la figure 2), les effectifs de produits sont très inégalement répartis. En effet si certaines catégories regroupent plusieurs centaines de références, d'autres n'en contiennent que deux ou trois... ce qui biaiserait inévitablement la classification.

Pour palier à ce problème, il est décidé de **fixer un effectif minimal** E_{min} de produits par catégorie. En ce qui concerne les catégories ne vérifiant pas ce critère, nous avons fait le choix de les fusionner avec les catégories "soeurs" et de les regrouper dans la catégorie parent. Un exemple de réorganisation est représenté en figure 3, les labels retenus pour la classification étant donc les catégories représentées en rouge.

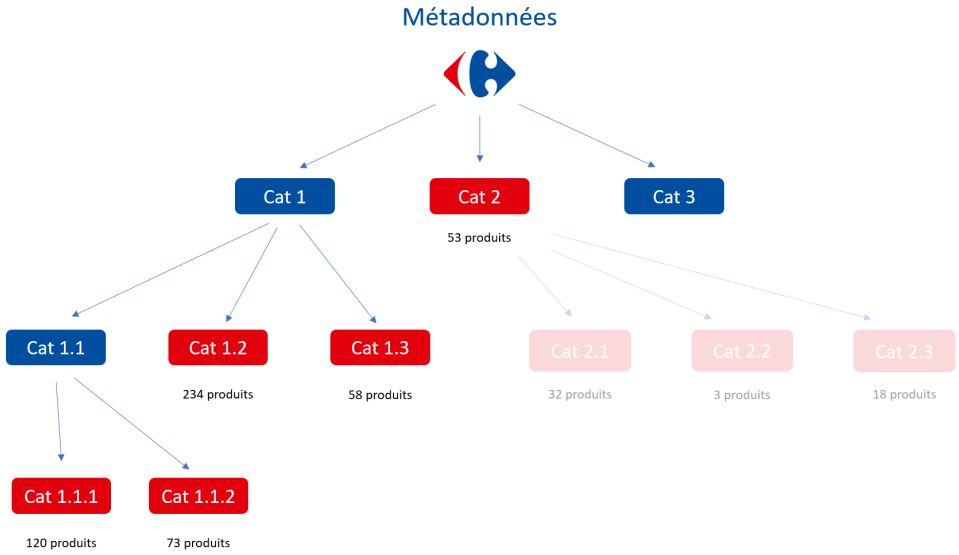

Figure 3 – Labels après transformations

1.2.2 • Determination de E_{min}

Ce choix de se ramener à des catégories introduit un nouveau paramètre : le nombre de produits minimal retenu pour définir la taille seuil moyenne des labels. En fonction de ce nombre, plus ou moins de catégories. Intuitivement, si il est exigé d'avoir beaucoup de produits dans chaque catégorie, l'agrégation des produits dans une catégorie parent et forcée et peu de catégorie *label* seront définies. On aura alors certes une bonne précision sur ces catégories, mais ces dernières seront très générales : "glaces", "tout pour une fêtes réussis", "viandes et charcuterie", "légumes". D'un autre côté, si trop peu de produits sont imposé par catégories, beaucoup de catégories seront utilisé pour la supervision de l'apprentissage mais les prédictions seront de qualité moindre avec une prédiction moins certaine.

Pour le problème présent, il est donc judicieux d'identifier **la bonne granularité** à utiliser. Dans le cadre de ce projet, avec l'ensemble de données à disposition, il a été choisi d'avoir au moins cinquante produits par catégorie standard. Ceci afin d'assurer un modèle suffisamment précis et un nombre de catégorie assez grand pour conserver un prédiction qui créer de la valeur-ajoutée. Ce pré-traitement des données est implémenté dans les fichiers : `metata_preprocessing.py` contenant les classes respectives au nettoyage et à cette tâche, et dans `metadata_main.py`, fichier exécutant les étapes successives requises. Ces fichiers sont trouvables dans le dossier git en ligne du projet : `github.com/bison-fute/Carrefour-Image-Recognition-Cha`

La détermination précise de ce paramètre ne se limite cependant pas à l'aspect purement data de ce challenge. Une étude marketing doit être réalisée afin d'englober tous les contraintes du problème et de Carrefour. Cette étude, dépassant le cadre de data, est laissée aux équipes de l'industriel partenaire.

1.3 Pre-processing des images

Un autre aspect important de ce problème était de parvenir à reconnaître un produit peu importe l'angle de prise de vue, la luminosité de la pièce, ou encore le bruit de l'image. Pour permettre au modèle reconnaître un produit malgré ces facteurs aléatoires, l'application de **transformations aléatoires** à notre set d'image lors du chargement dans le modèle a été mise en place. Un exemple de transformation des images est représenté en figure 4.

Enfin, pour préparer la phase suivante, il est choisi de normaliser toutes les images, en les transformant au format 224×224 pixels.

Figure 4 – Exemples d'images transformées

2
MODÈLES RETENUS

2.1 Transfer Learning

Le problème à traiter était un problème de classification d'images. Étant donnée la complexité d'une telle tâche et les ressources informatiques nécessaires pour entraîner un réseau de neurones sur un dataset aussi grand, nous avons fait le choix d'adopter une **approche par transfer learning**. Un réseau de neurones déjà entraîné à traiter un problème similaire de classification d'images a donc été réutilisé.

Dans la pratique, on n'utilise pas l'ensemble des couches du réseau pré-entraîné. En effet, les dernières couches d'un tel réseau répondent à un problème de classification spécifique, avec des catégories différentes de celle de notre problème. On appelle ces couches de classification le **head**. En revanche les premières couches du réseau, qui servent à extraire l'ensemble de caractéristiques d'une image, peuvent être réutilisées. On les appelle le **base**.

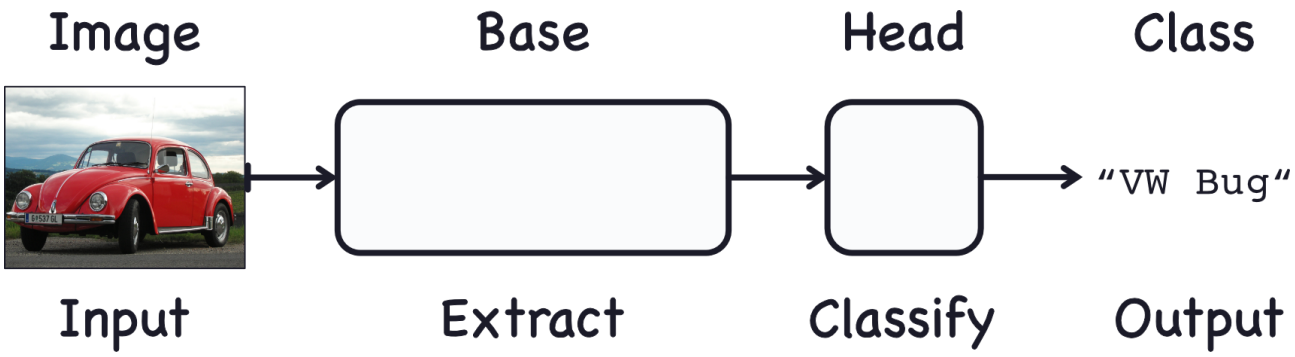

Figure 5 – Structure Head Base

Ainsi, le principe est d'utiliser le base d'un réseau pré-entraîné et de créer le head qui répondra à la problématique de classification. On présente dans les parties suivantes trois architectures typiques adaptées à la problématique du challenge : VGG-16, DenseNet161 et ResNet50. Elles ont été testées et comparées afin de sélectionner la plus performantes pour la tâche donnée.

2.2 VGG-16

VGG16 est un modèle de réseau neuronal convolutif proposé par l'équipe Visual Geometry Group de l'Université d'Oxford [1]. Le modèle atteint une précision de 92,7 % dans le test top-5 d'ImageNet, qui est un ensemble de données de plus de 14 millions d'images appartenant à 1000 classes. C'est l'un des célèbres modèles soumis à l'ILSVRC-2014. Il utilise massivement des convolutions avec des filtres de tailles 3x3. VGG16 a été entraîné pendant des semaines et a utilisé des GPU NVIDIA Titan Black. La démarche ici est donc de entraîner une dernière couches associée aux classes que l'on souhaite prédire.

2.3 DenseNet161

Le DenseNet connecte chaque couche à toutes les autres couches de manière prospective. Alors que les réseaux convolutifs traditionnels à L couches ont L connexions - une entre chaque couche et la couche suivante - le réseau d'intérêt réseau possède L(L+1)/2 connexions directes. Pour chaque couche, les cartes de caractéristiques de toutes les couches précédentes sont utilisées comme entrées, et ses propres cartes de caractéristiques sont utilisées comme entrées dans toutes les couches suivantes. Les réseaux denses présentent plusieurs avantages convaincants : améliorent la propagation des paramètres par descente de gradient, encouragent la réutilisation des caractéristiques et réduisent considérablement le nombre de paramètres. Les DenseNets apportent des améliorations significatives par rapport à l'état de l'art sur la plupart d'entre elles, tout en nécessitant moins de calculs pour atteindre des performances élevées. Cette architecture a été établie et pré-entrainé sur ImageNet par Huang en 2016 [2] et utilisée dans ce projet à l'aide du transfert learning.

2.4 ResNet50

Les Resnets sont une sorte de convolutional neural networks appelés Residual Networks. Ils sont très profonds par rapport à un réseau comme VGG, ResNet50 [3] possède 50 couches contre 16 pour le premier. Les resnets introduisent des connexions résiduelles entre les couches, ce qui signifie que la sortie d'une couche est une convolution de son entrée plus son entrée. De la même façon que pour VGG16, on ré-entraine une dernière couche de fully-connected pour adapter le réseau à notre problème et ses classes.

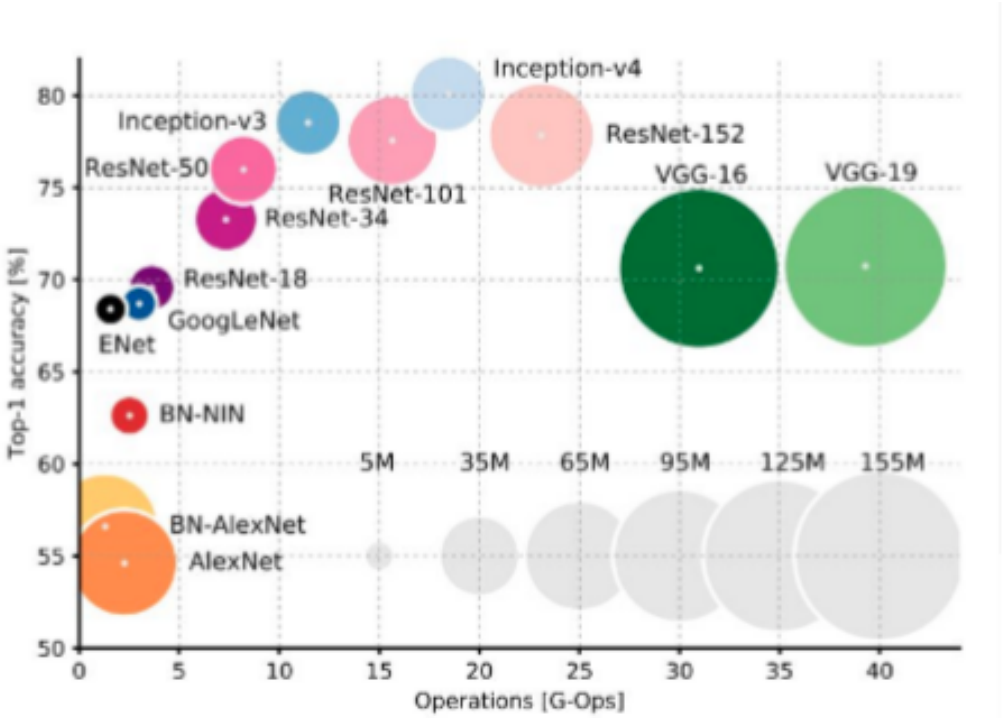

Figure 6 – Comparatif des performances de différents réseaux sur le dataset Imagenet

3
RÉSULTATS

3.1 Comparaisons des modèles

Les résultats de ce travail sont présentés dans la section suivante. Le dataset complet se compose des trois dossier parents `part1`, `part2` et `part5`. Les trois dossier totalise plus de 110Go de données, le choix est fait de n'utiliser que les données contenues dans `part5`, dossier de **15Go mieux proportionné à la puissance de calcul** à notre disposition. Pour quatre époques d'entraînement, les modèles prenaient en moyenne deux heures à s'entraîner.

La majorité des calculs ont été réalisés sur une machine au spécifités suivantes : DELL Inspiron 15 5000, GPU Intel Core i7-6500U, RAM 16 Go, GPU Intel HD Graphics 520. L'execution du code a lieu dans le fichier `main.py` trouvable dans le dossier git en ligne du projet : `github.com/bison-fute/Carrefour-Image-Recognition-Challenge`.

Modèle	**VGG-16**	**DenseNet161**	**ResNet50**
Précision sur l'ensemble de test	0.36	0.45	0.36

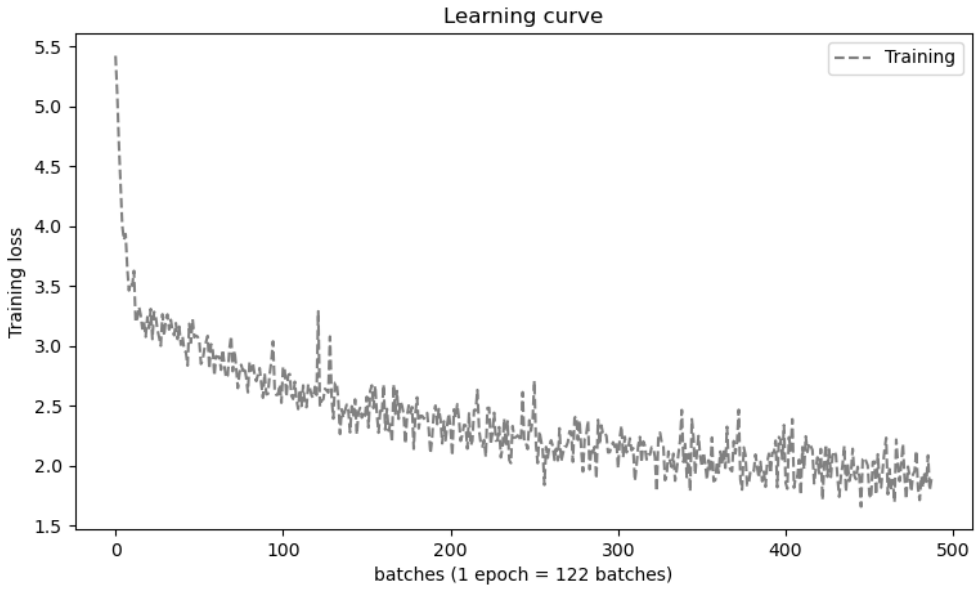

Figure 7 – Courbe d'apprentissage de l'architecture Denseet161

La courbe d'apprentissage du modèle DenseNet161 adapté à la problématique cible est donnée en figure 7. On peut notamment voir que la courbe d'erreur continue de chuter et que

l'apprentissage aurait pu être continuer. D'autre part, on vérifie que l'erreur d'entraînement est quasiment égale à l'erreur de validation ce qui confirme que le modèle n'a pas sur-appris l'ensemble de données.

Le modèle retenu pour cette étude est donc le **réseau pré-entraîné DenseNet161** sur le dataset d'ImageNet, affichant une précision de 0.45 sur notre ensemble de test du dossier de données `part5`. Une information supplémentaire intéressante à connaître est le rang moyen de la véritable catégorie cible parmi les catégories les plus probables données par l'architecture mise en place. La figure 8 rend compte la distribution du rang de la catégorie cible prédit par le modèle.

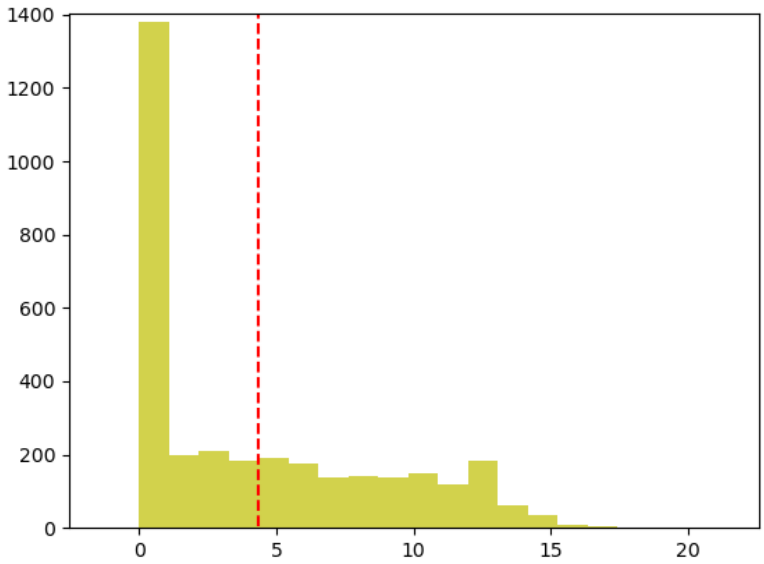

Figure 8 – Histogramme du rang de la vraie catégorie parmi les prédictions en sortie du modèle, en rouge la valeur moyenne.

3.2 Pistes d'améliorations vers l'objectif initial

Sont détaillées ici quelques pistes d'amélioration qui pourrait permettre d'atteindre les deux objectif définis : la reconnaissance précise d'un produit à partir d'une image, l'identification d'une catégorie,.

3.2.1 • Augmentation de données

Comme évoqué dans la partie I, il est assez difficile d'entraîner un réseau de neurones à partir des seulement 4 à 6 images à disposition par produit. Pour atteindre un tel niveau de précision, il faudrait disposer d'un plus grand nombre d'images par produits, et l'augmentation de données pourrait être appliquée si une plus grande puissance de calcul était à disposition.

La réalisation d'une telle augmentation consiste ici à **reprendre le code du projet autour des transformation aléatoires** appliquées aux images, de les multipliées et de les conserver dans nos données d'entraînement. On multiplierais ainsi artificiellement la taille de notre ensemble de données. Cette technique est massivement utilisée et à prouver son intérêt, elle est d'ailleurs également exploité à travers avec les convolutions ayant n (>1) noyaux/filtres, correspondant à une n-ième nouvelle version d'un image circulant dans le réseau.

3.2.2 • Pré-calcul des features

Les poids du base de l'architecture pré-entrainée sont figés. Ceux-ci ne sont pas modifiés lors de l'entraînement du réseau. Le passage des données dans ce réseau est coûteux car beaucoup d'opérations sont effectuées afin d'extraire les caractéristiques des images. De plus, les premières couches n'étant pas entraînées, celles-ci effectueront toujours les mêmes calculs et leur sortie sera toujours identique. Ainsi, il est intéressant de **faire passer une unique fois les données dans le base, puis d'entraîner ensuite le head** autant d'époque qu'on le veut. Ayant utilisé relativement peu d'époques lors de l'entraînement, la démarche n'a pas été mise en place. Toutefois, avec l'utilisation d'un plus grand dataset, et un entraînement comprenant plus d'époques, cette procédure aurait été indispensable.

4
CONCLUSION

Ce projet de reconnaissance d'image a été mené conjointement par Carrefour dans le cadre de ce cours de mathématiques appliquées. Le but de ce projet est d'arriver à reconnaître à partir d'une photo d'un produit prise par un consommateur le produit du catalogue Carrefour associé. Ce premier objectif a été identifier comme irréalisable dans le cadre d'une tâche d'apprentissage supervisé à partir de l'ensemble de données à disposition. Pour répondre à celui-ci, une méthodologie plus évidente peut plus facilement être mise en place grâce au code-barres des produits : l'utilisateur pourrait alors simplement *scanner* ce code et identifier le produit.

Afin de tirer profit des données utilisées pour ce challenge, un deuxième objectif a été défini : celui de prédire les produits similaires à partir de l'image, la vision d'un produit. Ainsi, l'application business devient celle de, à partir d'une image d'un produit, *proposer un produit similaire ou un substitut au consommateur.* Dès lors, l'objectif était de définir comment savoir ce qu'est un produit similaire : est-ce un produit de la même couleur, un produit du même industriel, le même type de produit, etc. Pour ce faire, l'utilisation est faite des meta-données des produits de Carrefour, on choisit d'agréger les produits par catégories similaires : "viandes", "produit transformé à base de viande", etc.

A ce stade, il s'agit alors de mener une étude data de la problématique : nous cherchons à prédire une catégorie de produit. La catégorie doit comporter suffisamment de produits pour que l'entraînement et la prédiction par un méthode de machine learning soit rentable, tout en identifiant des catégories d'une granularité adapté et suffisamment fine : des produits très similaires enclin à être utiles au consommateur. Le pre-processing des données a donc représenté une part majoritaire du travail sur ce challenge et un algorithme de pre-processing a été défini à cette fin précédemment introduite. La taille minimale moyenne par catégorie est laissée en paramètre et pourra être ajusté par les équipes à l'interface data et marketing du partenaire de ce challenge.

La quantité de données a été difficile à traiter avec les ordinateurs de particulier à notre disposition. Le traitement d'un tel amas aurait été plus adapté avec des serveurs et des services de cloud computing qui n'étaient malheureusement pas disponibles. Afin de palier à ce problème, le choix a été fait de se concentré sur une partie restreinte de l'ensemble de données, suffisamment représentatif et variés pour être étendu à plus de données.

Les résultats obtenus sont encourageants : une précision de 0.45 pour un ensemble images-catégories cibles ayant 240 classes en sortie. Pour information et à titre de référence, une prédiction aléatoire sur ces catégories aurait donné une prédiction de 1/240 soit une précision de 0.0042 de deux ordres de grandeur de différences. On montre également que la catégorie cible est en sortie du modèle parmi les 5 premières catégories en moyenne et majoritairement correcte.

RÉFÉRENCES

[1] Karen Simonyan et Andrew Zisserman : Very deep convolutional networks for large-scale image recognition, 2015.

[2] Gao Huang, Zhuang Liu et Kilian Q. Weinberger : Densely connected convolutional networks. CoRR, abs/1608.06993, 2016.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren et Jian Sun : Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015.