HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Enzo Allemano
Yohann Paulus

17/06/2022

Perez Uribe Andres
Izadmehr Yasaman

Apprentissage par Réseaux de Neurones artificiels

Rapport laboratoire 5:
Transfer learning

&

Object recognition in the wild using Convolutional Neural Networks

Introduction

L'application que nous développerons permettra de différentier des types de câbles courants, utiliser dans le domaine de l'informatique.

Cela peut être utile pour le rangement de l'inventaire d'un service informatique.

La plupart des photos sont récupérées via le package « bing-image-downloader ». Cependant, nous prendrons également nous-même des photos des types de câbles cités ci-dessous, dans le but de diversifier davantage notre dataset. Nous aurons donc des photos d'un même câble pris sous différents angles dans différents endroits pour également diversifier l'arrière-plan.

Ce travail sera réalisé grâce à CNN et au transfer learning.

The problem

Nous allons ici essayer de différencier quatre types de connecteurs.

Respectivement :

- RJ-45
- HDMI
- VGA
- IEC-320

La database collectée est l'ensemble des images récupérées sur internet et de nos photos. Nous essayons ici de créer un small dataset dit « Balanced ». Nous aurons donc :

- 50 photos de RJ-45 + photos persos
- 50 photos de HDMI + photos persos
- 50 photos de VGA + photos persos
- 50 photos de IEC-320 + photos persos

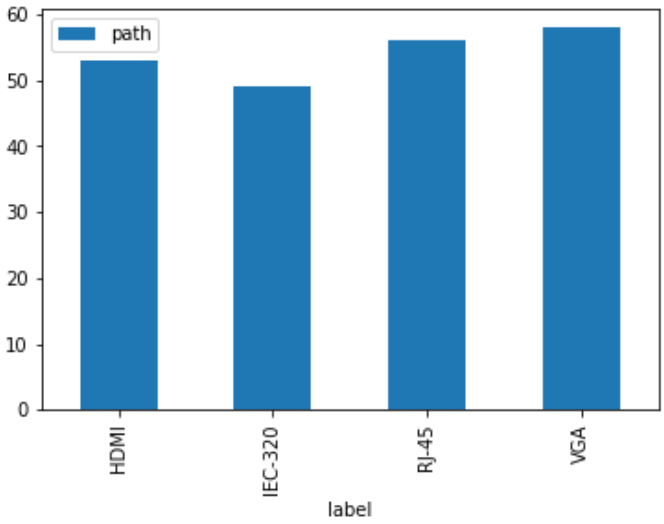


Figure 1 Number of images per class

Inter-class similarity

VGA

Pour les VGA le coefficient de similarité est relativement faible. En effet, nous avons beaucoup de vues différentes pour un même connecteur

HDMI

Concernant les connecteurs HDMI, on peut voir qu'on obtient un coefficient de similarité relativement bon. En effet, l'échelle et les prises de vues sont souvent très similaire.

RJ-45

Ici, notre coefficient de similarité est plutôt moyen. Les prises de vues sont plus ou moins les mêmes mais certaines fois, des objets sont rajoutés à la photo.

IEC-320

Pour ce type de connecteur le problème est qu'il peut prendre différente forme. C'est donc un coefficient de similarité mauvais. Ici, il va clairement falloir faire de la préparation de données.

Data preparation

Le pré-processus de préparation des données c'est déroulé sans redimensionnement des images et sans normalisation. Cependant, les images ont dû être triées en vue de ne garder que celles jugées pertinentes pour entraîner notre modèle. Nous avons donc supprimé à la main la plupart des images qui contenaient des réplicas de connecteurs en une seule image et celles qui montraient plus qu'un connecteur.

Séparation du train dataset, validation dataset et test dataset

Pour la séparation de tous les datasets, nous avons utilisé la fonction « `train_test_split` » du package « `sklearn.model_selection` » qui nous permet de tirer un test dataset au hasard en lui spécifiant une taille. La taille du dataset représente 20% du datasets entier.

Le train dataset est quant à lui créé grâce à la fonction « `create_dataframe_from_directories` » déclarée dans la première cellule du notebook.

Model creation

Comme nous avons pu le constater le choix des hyperparamètres sont cruciaux pour obtenir un bon modèle. Nous nous sommes donc basés sur les différents résultats obtenus pour chaque changement de ces paramètres.

Pour converger vers notre modèle final nous avons analysé les hyperparamètres suivants :

- nb epochs = 4
- dense = 0.5
- batch size = 32
- Dropout = 0.3

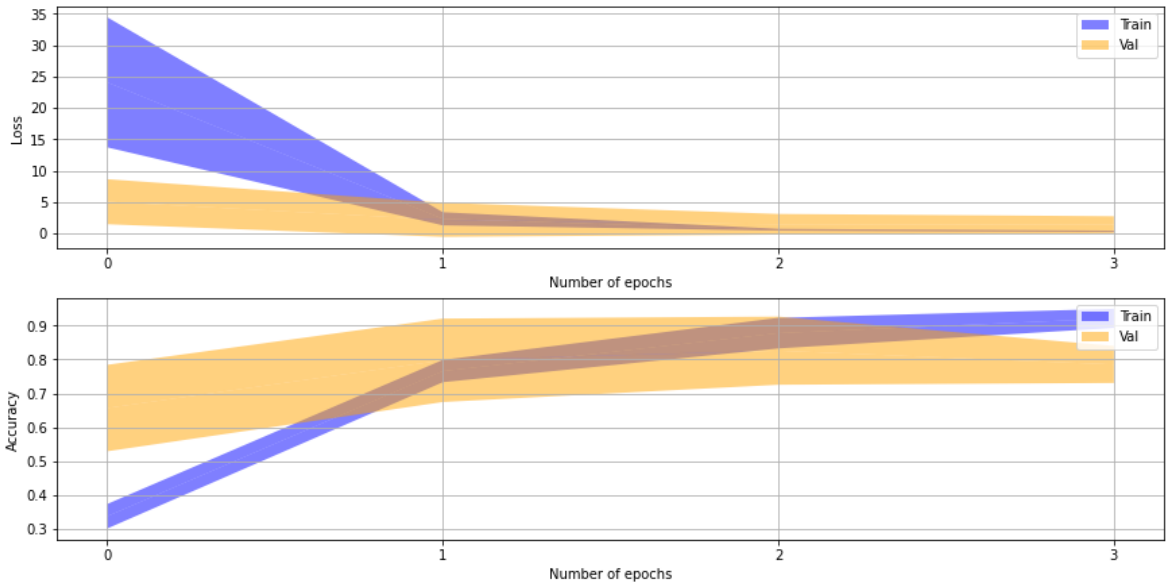
```
 Layer (type)                Output Shape              Param #     Connected to
==================================================================================================
 input_15 (InputLayer)       [(None, 224, 224, 3  0           []
                             )]

 Conv1 (Conv2D)              (None, 112, 112, 32  864         ['input_15[0][0]']
                             )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128      ['Conv1[0][0]']
                             )

 Conv1_relu (ReLU)           (None, 112, 112, 32  0           ['bn_Conv1[0][0]']
                             )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288     ['Conv1_relu[0][0]']
 wiseConv2D)                 )

 expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128     ['expanded_conv_depthwise[0][0]']
 tchNormalization)           )

 expanded_conv_depthwise_relu (  (None, 112, 112, 32  0       ['expanded_conv_depthwise_BN[0][0
 ReLU)                       )                                ]']

...
Total params: 5,394,238
Trainable params: 3,136,254
Non-trainable params: 2,257,984
```

L'architecture de notre modèle est relativement simple et commune. Concernant le transfer learning, celui-ci a été réaliser avec « `mobilenet` ». Le transfer learning est important dans notre modèle d'apprentissage car nous n'avons pas de grand dataset et la puissance de calcul nécessaire. Cela nous aide particulièrement bien car les images de connecteurs qui composent notre dataset ne sont pas réellement normalisées. Il faut donc être capable de reconnaître un connecteurs sous tous ses angles et sous toutes ses formes. Ainsi, nous augmentons la capacité de notre modèle d'apprentissage composé du transfer learning à reconnaître nos connecteurs.

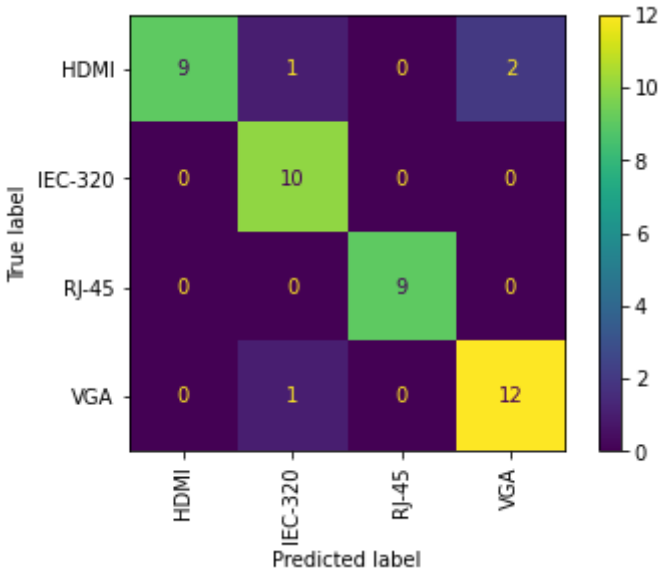
Results

a)

Plot :



Matrice de confusion :



b)

F1-score :

HDMI : 0.8571

IEC-320 : 0.9524

RJ-45 : 1.0000

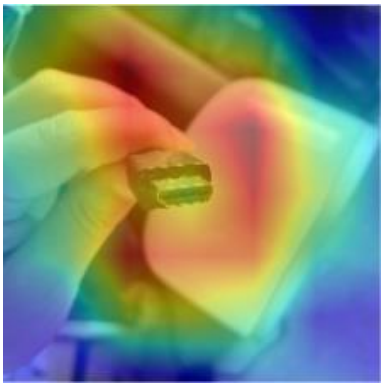
VGA : 0.8889

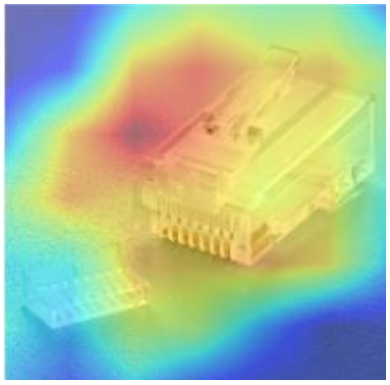
c)

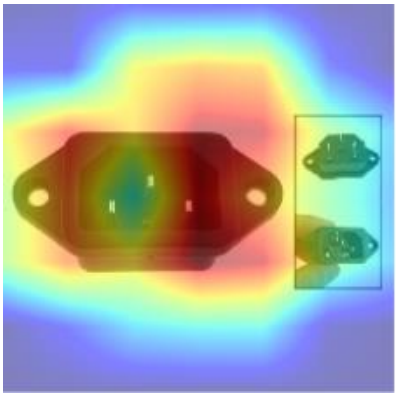
Comme on peut l'observer au point a), la précision de prédiction lors des tests surpasse légèrement la précision lors de l'entrainement. Lors de l'utilisation de l'application mobile, nous avons remarqué que notre modèle est très sensible à l'angle de vue qu'il a sur le connecteur. En effet, tourner un peu le connecteur peut faire varier la prédiction avec une certitude de plus de 90%. Par exemple un connecteur de type RJ45 vue de face est souvent reconnu comme un connecteur HDMI par notre modèle. Mais dès que l'on montre le connecteur de profil ou vu de dessus, le modèle le prédit correctement avec la même certitude. Cela vient sans doute des données d'entrainement car les connecteurs sont régulièrement pris en photos avec le même angle de vue. Par exemple les connecteurs HDMI sont souvent pris de face car c'est ce qui nous permet de les différencier avec d'autres connecteurs (Ex : Display Port), une vue du dessus nous intéresserais moins, voire pas du tout.

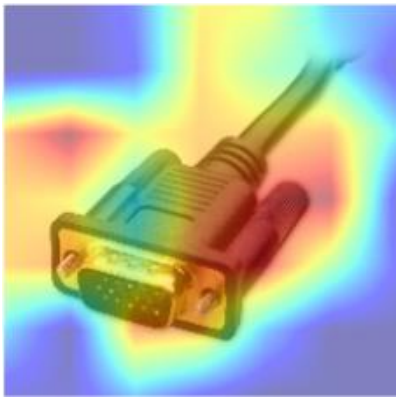
d)

On peut remarquer que notre modèle s'en sort principalement bien dans les quelques exemples si dessous :



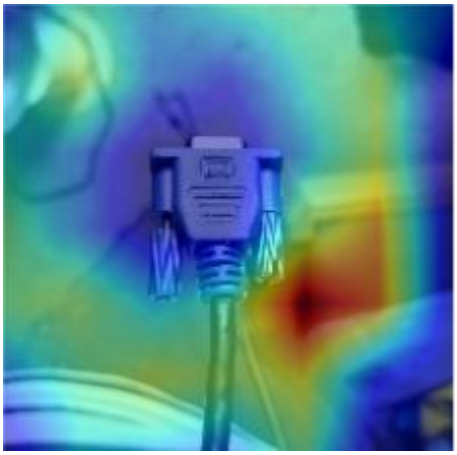
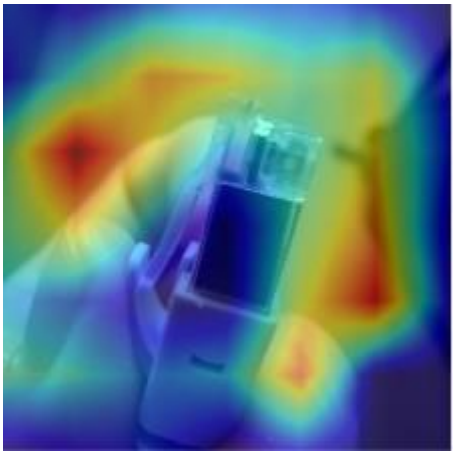
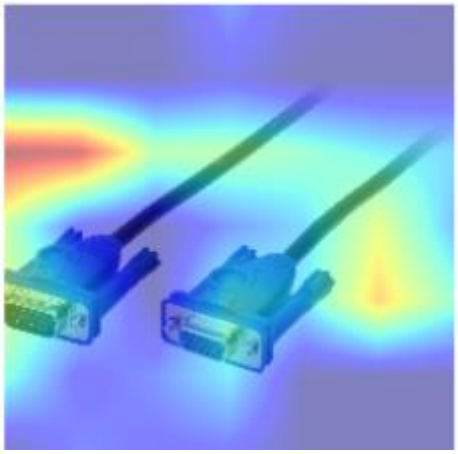






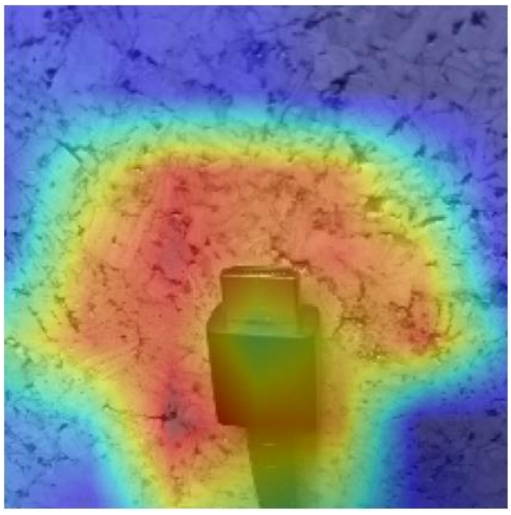
Cependant dans certain cas, il passe totalement à côté de l'information. Et cela pour des images que nous avons prises nous-mêmes comme pour des images tirées d'internet.

Exemple :

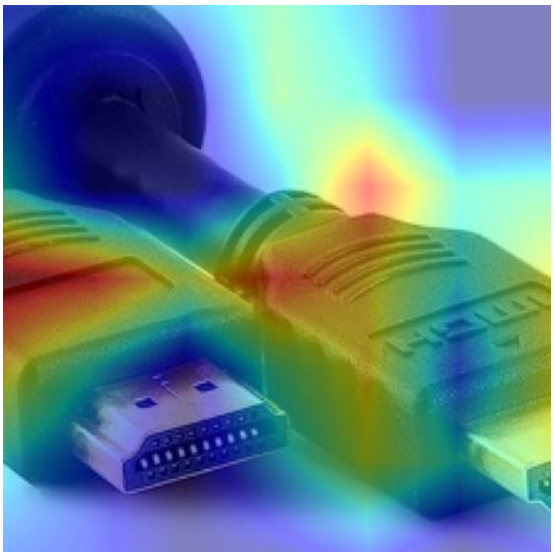





e)

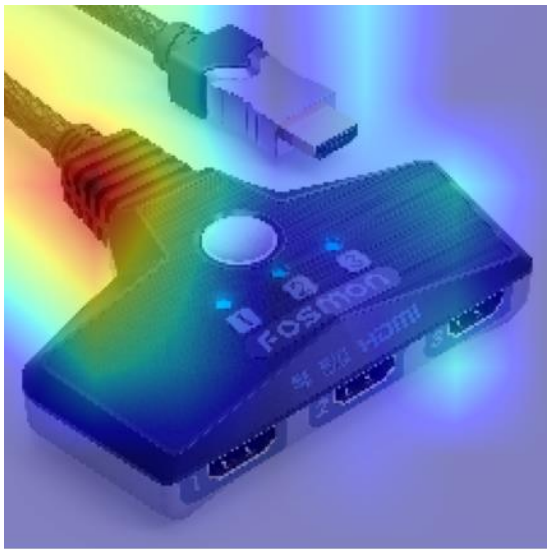
Dans cet exemple de mauvaise classification on peut supposer que l'angle de vue impact la décision du modèle. En effet comme mentionné plus haut, les connecteurs HDMI sont rarement orienté dans ce sens. Notre modèle a donc de la peine à reconnaître le connecteur.



Ici, le modèle a de la peine à détecter les points d'intérêt pour la classification. Le connecteur ne se situe pas au milieu de l'image. De plus il y en a deux sur l'image dont un partiellement coupé.



Ici on remarque avec la heat map qu'il ne détecte aucun des 4 connecteurs.



Ici on peut remarquer que le connecteur est bien détecté malgré l'image en bas à droite. Cependant la forme particulière du connecteur induit notre modèle en erreur.

f)

Avec plus d'image et plus d'angle de vue sur les différents connecteurs comme expliqué plus haut. Prendre plus d'image nous-même avec différents fonds aiderait aussi le modèle pour une utilisation plus proche d'une utilisation réelle. Des images avec un seul câble et un seul connecteur serait de meilleure qualité et plus pertinente car en réalité lorsque nous utilisons l'application nous lui montrons uniquement un seul connecteur.

g)

La classe HDMI est celle qui pose le plus de problème. Pas spécialement, c'est difficile de trouver des images d'un connecteur HDMI sous tous les angles possibles. De plus il en existe de différentes formes et nos images d'entrainement contiennent des hubs, plusieurs connecteurs sur une image, … Ce qui pose un problème pour l'apprentissage. Lorsque l'on tente de scanner des connecteurs qui ne font pas partis du modèle, ils sont généralement classifiés comme étant de type HDMI. Si les connecteurs ont des similarités avec un connecteur du modèle ils seront alors reconnus comme ce dernier mais avec un pourcentage de certitude assez faible (60% - 70% environ). Par exemple un connecteur DVI, très similaire à VGA, sera reconnu comme tel. Pour un RJ-10, il sera perçu comme un RJ-45. Nous avons remarqué que le fond de l'image n'influençait pas spécialement lors de la prédiction. Cela provient sans doute ne notre volontés à prendre des photos avec un fond différent pour nos données d'entraînements.

Conclusions

Pour conclure ce rapport, il était facilement possible d'effectuer du transfer learning avec MobileNet, tous cela en seulement une centaine de lignes de codes.

Nous avons obtenu un modèle de classification qui répond partiellement à nos attentes avec seulement ~200 photos. Cependant, les limites de notre modèles sont relativement vites atteintes. Effectivement, comme mentionné plus haut, l'orientation du connecteur influe fortement la prédiction.

Par exemple, il n'est actuellement pas possible de prendre une photo contenant plusieurs connecteurs et de faire différencier à notre modèle d'apprentissage.

Une amélioration possible pour notre modèle d'apprentissage serait par exemple de lui faire reconnaître « aucun » câble. De cette manière, il n'affichera pas de résultat 100% erronés alors que nous sommes en train de lui présenter un autre objet quelconque.

Il serait aussi possible d'augmenter le nombre d'image d'entrainement avec des images de connecteurs sous tous les angles de vues. Cela permettrait de réduire l'impact de l'angle de vue sur la prédiction.

Ce dernier laboratoire nous a permis de mettre en pratique tous les éléments de théories vu en cours en un seul rendu. Même si notre modèle d'entraînement n'est pas exceptionnel, cela a été très enrichissant de pourvoir un construire un de « A à Z ».