Master 1 IMAGINE

GASC Thibault
DIAB Ingo

Compte rendu 5 Projet : Détection de zones copiées-déplacées dans des images

Année Universitaire 2022-2023

Table des matières

I Travail de la semaine

1 Partie sans CNN

Cette semaine nous avons montré nos résultats et on s'est rendu compte qu'il y avait un "problème" sur certaines images lors de la détection. En effet, sur certaines images, notamment celles avec des écritures, lors de la détection, il est détecter qu'il y a une falsification au niveau de ces textes même si ce n'est pas le cas.

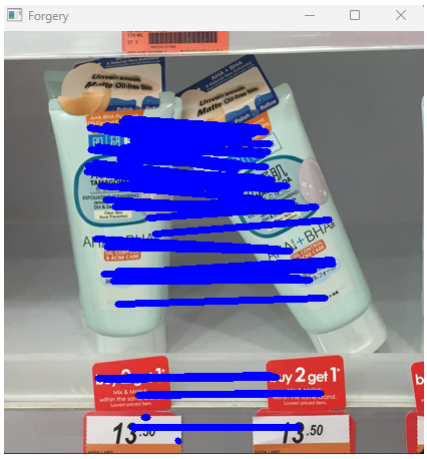

FIGURE 1 – Image issue de la détection

Voici un exemple où il y a ce problème. La zone ou il y a le prix a été détectée comme étant falsifiée, cependant elle ne l'est pas.

Lors de la dernière séance avec Mme. Jansen Van Rensburg nous a dit qu'il fallait prendre en compte le bruit sur les images. Si le bruit sur les deux éléments est le même alors c'est qu'il y a eu un copié-déplacé. C'est donc ce que l'on va essayer de faire pour la semaine à venir.

Un autre problème rencontré est celui de la taille des images. Les paramètres de DBSCAN dépende de la taille de l'image, en effet, si on prend une image 512×512 et que l'on applique DBSCAN avec des paramètres défini et qu'après on change d'image et que l'image ne fait plus 512×512 mais 275×183

2 Partie CNN

Nous avons trouvé un moyen d'utiliser l'intégralité de notre dataset sans surcharger la RAM. Pour cela nous utilisons un Generator. Ce générator va,

à partir d'un dossier contenant les images classées selon leur label, remplir des batchs d'une certaine size.

Nous avons donc, au préalable, trié nos images selon Original ou Forged et les avons regroupé dans le dossier Dataset/ParsedImages/0 pour les images Originales et Dataset/ParsedImages/1 pour les images Forged. Nous utilisons ensuite ImageDataGenerator avec un "validation_split" de 30%. A partir de ce ImageDataGenerator, nous pouvons utiliser la méthode flow_from_directory(). Cette méthode prend en paramètre, entre autre, le path du dossier contenant les classes (Dataset/ParsedImages) et le nombre d'images à générer (batch_size).

Nous avons un batch_size à 32 donc seulement 32 images seront chargées en même temps, ce qui fait que notre RAM n'augmente que d'environ 1/1.5 Go et nous pouvons donc entraîner notre modèle sur 10 000 images sans surcharger la RAM.

Nous pouvons en plus utiliser des effets de translations, rotations, zoom, ... sur les images chargées (en donnant les effets voulu à notre ImageDataGenerator) ce qui permet de faire en sorte que notre modèle n'apprenne pas par coeur.

Nous avons aussi rajouter la prise en considération d'autres mesures telles que le Recall et la Precision (pour ensuite calculer le F1-Score).