



Master 1 IMAGINE

GASC Thibault

DIAB Ingo

Compte rendu 3 Projet : Détection de zones copiées-déplacées dans des images



Année Universitaire 2022-2023

Table des matières

I	Résumé de la semaine	2
II	Détection de points clés	2
III	Interface graphique	3
IV	CNN	4

I Résumé de la semaine

Cette semaine, nous avons commencé la partie code du projet. On a choisit de travailler avec le langage Python, car c'est le plus simple pour utiliser des bibliothèques telle que OpenCV ou pour faire des réseaux de neurones. On s'est divisé les tâches de la manière suivante : Ingo travaille sur la partie réseaux de neurones et moi sur la partie détection de points clés.

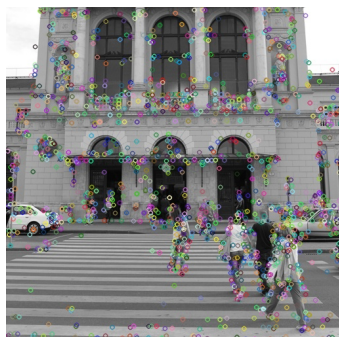
Lors du cours de mardi de cette semaine (semaine du 06/03), Mme. Jansen van Rensburg est passée nous voir pour savoir comment nous allons aborder le sujet. On lui a donc expliqué ce qui est mentionné plus haut et elle nous a conseillé de ne pas perdre de temps sur la partie réseaux de neurones et de se focaliser plus sur la partie détection, sauf si on se sent à l'aise comme cela. Etant plutôt confiant, on a donc décidé de commencer le projet comme nous l'avions initialement prévu quitte à, si on voit que l'on avance pas assez vite, se mettre tous les deux sur la partie la plus importante (la partie sans CNN).

II Détection de points clés

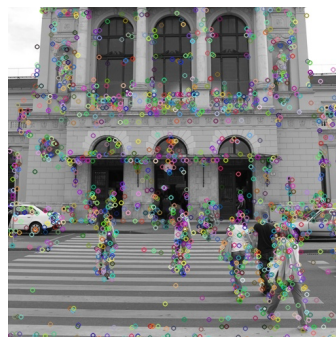
Pour la partie détection, on a décidé d'utiliser l'algorithme SIFT (Scale-Invariant Feature Transform) car selon les images il peut y avoir des changements d'échelles, des rotations, des changements de point de vue et SIFT peut détecter des caractéristiques qui sont invariantes à tout ça.

L'algorithme SIFT permet de trouver les points clés dans une image en utilisant des techniques de détection de bords/coin et en utilisant du seuillage. Après avoir fait cela, il produira un descripteur (vecteur de caractéristiques) pour chaque point clés.

Puisque nous travaillons sur des images qui ont subi des copiées-déplacées, certaines parties de l'image auront leurs descripteurs de caractéristiques quasiment identiques, c'est comme cela qu'on détectera la falsification.



(a) image originale



(b) image falsifiée

III Interface graphique

Il est demandé de créer une interface pour pouvoir visualiser les résultats obtenus. Nous avons donc choisis d'utiliser PyQt car il plutôt simple à prendre en main et il est relativement simple d'importer les bibliothèques que l'on a besoin. Voici un brève aperçu de l'interface que l'on a pour l'instant.



FIGURE 2 – Interface avec PyQt

IV CNN

Nous avons commencé à utiliser le dataset CoMoFoD_small_v2. Dans un premier temps, nous avons chargé les images à l'aide de Numpy. Ce dataset ayant été créé pour être utilisé comme dataset d'un CNN, les images sont déjà pré-traitées (pas besoin de sur échantillonner/sous échantillonner une classe, les images ont toutes la même taille,...).

Une fois ces images chargées, nous les avons classifiées en 2 classes distinctes : Original et Forgery (originales et falsifiées) à l'aide de leur nom "XXX_O..." nous indique que l'image XXX est originale tandis que "XXX_F..." nous indique qu'il s'agit d'une forgery. Après avoir attribué aux images 0 (original) ou 1 (forgery) selon leur nom, nous allons ranger toutes les images dans un tableau et tous les labels dans un autre. Nous allons les mélanger afin que tout soit ordonné de manière aléatoire.

La dernière étape de préparation du dataset est la normalisation (diviser les images par 255).

Un problème est survenu lors de la préparation des données. La taille du dataset est bien trop importante, la RAM est surchargée et l'ordinateur est obligé de redémarrer. Pour palier à ce problème, nous allons utiliser Google Colab, ce qui nous permet de minimiser l'impact d'une RAM surchargée ainsi que d'avoir accès à un GPU puissant (pour entraîner le modèle). Nous comptons ensuite exporter le modèle quand il sera entraîné afin de le rendre utilisable à partir de notre interface graphique.

Le problème de RAM n'étant pas réglé, nous prévoyons de diminuer la taille du dataset utilisé quitte à l'augmenter de nouveau si le modèle n'est pas assez performant.

Lien du notebook :

<https://colab.research.google.com/drive/1JZZSSggbLRODCWJhlaoCUNwG6dkM7rBD>