

Rendu TP Visage

Boudo Louis 3410483

January 15, 2021

Introduction

À travers ce TP, nous allons essayer et exposer la méthode détection de visage avec les points caractéristiques. Le lien du git : <https://github.com/b-louis/Visages>

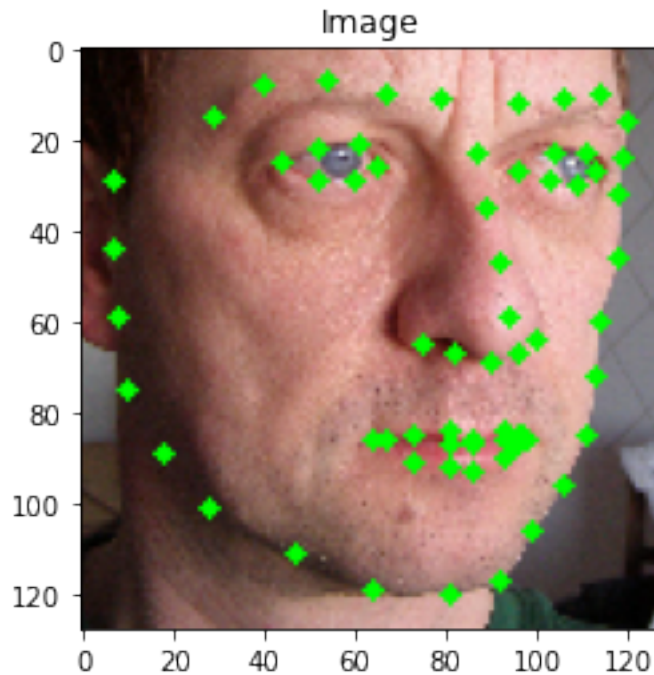


Figure 1: Exemple d'image d'entraînement avec les annotations

1 Préparation des données

1.1 Visualisation des données

En observant les images de la base, on constate que d'une part, ces dernières sont hétéroclites tant par leur prise de vue que leur qualité. Et d'autre part, elles ont toute le visage au centre de l'image.

1.2 Augmentation des données

Ajouter ce bruit, permet d'enrichir la base d'entraînement avec des exemple représentant des "nouveaux cas". Le but est de rendre l'algorithme plus performant et plus généralisable (On s'imagine par exemple le cas d'annotations erronées). Pour trouver l'amplitude de ces déplacements, on pourrait faire la différence entre la valeur terrain et ces valeurs modifiées.

2 Apprentissage d'un régresser simple

2.1 Extraction de caractéristiques image

1. La valeur de chaque pixel nous donnerais d'une part, pas assez d'information (exemple : on pourrait, confondre un pixel clair des dents et du blanc des yeux) et d'autre part, cela ferait énormément de paramètres (pour une image 100×100 cela fait 10 000 paramètres).
3. Les descripteurs sont de dimension 128, cela correspond à $4 \times 4 \times 8$ soit 4 fenêtres composées elle même de 4 fenêtres plus petites avec 8 directions possible pour les vecteurs. Pour chaque images on en a 68.
4. En concaténant, on pour chaque image, un descripteur de taille 8704
5. A part du *SIFT*, on pourrait utiliser une représentation avec des histogrammes sur les points caractéristiques. Il y a aussi les descripteurs *ORB* qui peuvent être une bonne alternative aux *SIFT* en offrant une vitesse de calculs plus rapides et des performances aussi bonnes voir meilleures qu'avec *SIFT*.

2.2 Réduction de dimensionalité

1. Dans certains cas où, on a beaucoup de features il est trop compliqué et trop chère (informatiquement parlant) d'apprendre directement sur la base. En plus de ça, la robustesse de l'algorithme est compromise avec un modèle trop compliqué. Le fait de réduire le nombre de features permet de, simplifier le problème ce qui rend les calculs plus rapides, mais aussi d'obtenir un problème plus généralisable en se focalisant seulement sur les features les plus importantes.
3. 3091, on est passé de 8704 à 767 features. On a pu garder 98% de la variance en ayant seulement 8% des features. On observera par la suite que pour 767 features, les calculs sont conséquents, soit environ 30 minutes pour pour calculer les *SIFT* et \tilde{X} .

2.3 Estimation du déplacement

3. Pas grand chose sur l'efficacité de l'algorithme , si ce n'est que les points on été déplacés dans les directions les plus optimales, avec une erreur

d'environ 2.8 pixels pour cette première image.

2.4 Validation sur un ensemble de test externe

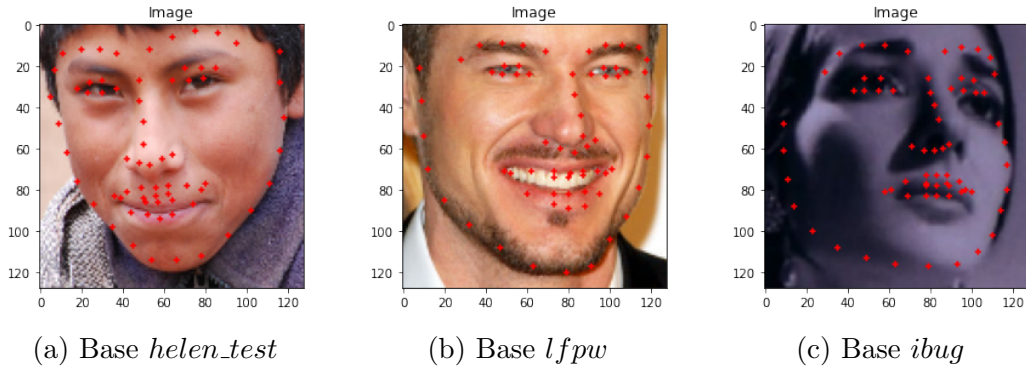


Figure 2: Résultats obtenus sur les bases de test avec les points du visages

1. Cela ne permet pas de pouvoir conclure quoi que ce soit car, on ne regarde que l'erreur d'une image qui est elle même dans la base de train sur laquelle on a appris . Pour avoir quelque chose d'exploitable, il faudrait regarder les performances sur une base de test.
2. On constate que les points sont déplacés dans la bonne direction, mais avec une erreur sur cet ensemble à 2.6 pixels.

3 Cascade de régresseurs

3.1 Itération du processus d'alignement

Dans les deux cas on constate peu de changements entre $k=1,2,3,4$.

1. On constate un certain raffinement entre $k = 0$ et $k \neq 0$, mais on note aussi que le raffinement fait perdre de la précision sur les contours du visage mais en fait gagner sur les autres contours (bouche, nez, yeux)
2. Sur les données test, les résultats sont similaire au cas d'entraînement cependant les résultats eux même dépendent surtout des images. En

effet, on aura plus de mal à trouver les bons points avec des visages où l'expression ou la pose est complexe (exemple la bouche ouverte ou un tête de 3/4 en contre plongée). On le voit clairement sur le set ibug.

3.2 Detection des points caractéristiques sur d'autres images

En effectuant quelques tests, on constate que l'algorithme n'est pas aussi performant sur des images pris sur internet. Le principal problème est la position du visage moyen de départ qui ne permet pas à l'algorithme de retrouver les points caractéristiques et donc d'effectuer les déplacements. Ensuite comme cité plus haut, l'algorithme a beaucoup de mal à restituer les visages avec des poses ou des expressions extrêmes.