Détection et reconnaissance faciale : De l'apprentissage automatique traditionnel à l'apprentissage profond

1 Inttroduction

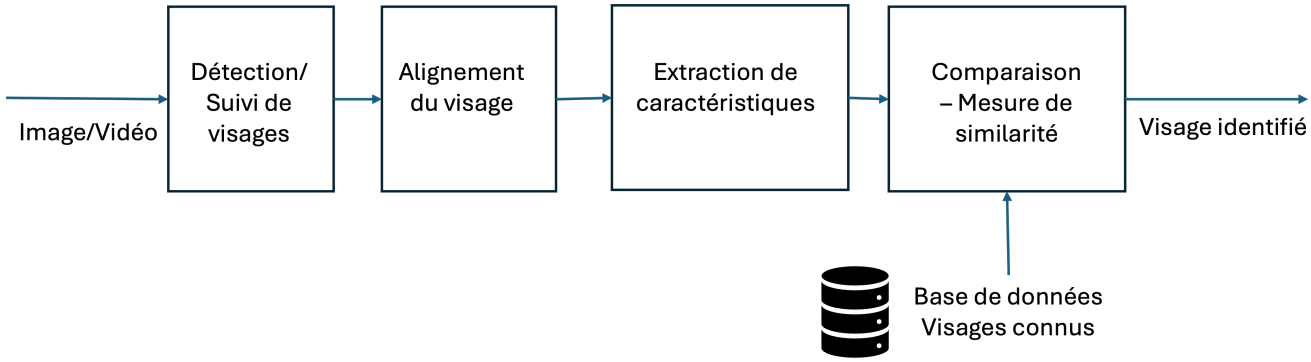

FIGURE 1 – Pipeline de reconnaissance de visage

La figure 1 présente les principales étapes d'un système de reconnaissance de visage à partir d'une image ou d'une vidéo. Ce système est composé en particulier de trois phases principales :

— Détection de visage
— Extraction de caractéristiques
— Reconnaissance de visages

La phase d'extraction des caractéristiques est une phase essentielle dans les approches classiques. Dans le cas des approches convolutionnelles, cette phase pourrait être remplacée par un CNN.

2 Détection de visages

La détection de visage est un cas particulier du problème de détection d'objets en vision par ordinateur où l'on cherche à détecter la présence et la localisation précise d'un ou plusieurs visages dans une image. Plusieurs approches ont été développées dans la littérature. En particulier, nous proposons d'utiliser la méthode de détection Cascade de Haar [**1**] (voir TP3).

2.1 Cascade de Haar

Cette méthode proposée par les chercheurs Paul Viola et Michael Jones en 2001 [**1**] a fait l'objet du TP3. Il s'agit de l'une des premières méthodes supervisées capables de détecter efficacement et en temps réel des objets dans une image. Proposée intialement pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures.
Etant une méthode d'apprentissage supervisée, la méthode de Viola et Jones nécessite une base d'apprentissage composée de plusieurs exemples de l'objet à détecter pour entraîner un classifieur. Une fois le classifieur est entrainé, il sera utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles.
Cette méthode est implémentée sous licence BSD dans Open CV et le modèle pré-entrainé pour la détection des visages pourrait être chargé directement sans refaire l'entrainement (voir TP 3).

2.2 Evaluation des performances d'une méthode de détection d'objets d'objets

Plusieurs métriques sont utilisées pour évaluer les performances des méthodes de détection d'objets. Ces métriques se basent essentiellment sur la métrique d'Intersection sur Union (IoU) qui quantifie à quel point la boîte englobante prédite par un algorithme de détection est correctement positionnée par rapport à la la boîte englobante réelle (vérité terrain). Elle permet également de quantifier les métriques précision et rappel des algorithmes de détection.

Intersection sur Union (IoU)

Elle est définie comme le rapport de la zone d'intersection à la zone d'union de la boîte englobante prédite et de la boîte englobante de la vérité terrain. L'IoU est un nombre compris entre 0 et 1. Un score IoU plus élevé implique une prédiction plus précise. En particulier, si IoU $= 0$, cela signifie que les deux boites englobantes coicident. Si IoU $= 1$, cela signifie qu'ils se chevauchent complètement.
Pour plus d'informations sur les métriques d'évaluation, vous pouvez consulter : Object Detection Metrics

2.3 Travail Demandé

1. Ecrire un programme Python permettant de détecter un visage à partir d'une image et d'un WebCam. Utiliser Cascade de Haar [**1**] (voir TP3).
2. Evaluer les performances de cette approche de détection sur un dataset de détection de visage public. Vous pouver utiliser le dataset suivant Kaggle Face Detection Dataset. (Obligatoire - Utiliser au moins la métrique IoU)
3. Interpréter les résultats obtenus.
4. **Bonus** : Vous pouvez utilser un autre détecteur de visage de votre choix (à décrire brièvement) et réaliser une comparaison entre les deux détecteurs.

3 Reconnaissance de visages

Une fois le visage est détecté, nous nous intéressons à le reconnaître. Dans cette section nous supposons qu'on dispose d'images de visages (c.a.d la phase de détection a été déjà réalisée). L'objectif est de reconnaître l'identité de chaque visage. Nous allons explorer des approches classiques et des approches CNN.

3.1 Approches classiques

Les approches classiques nécessitent une phase d'extraction de caractéristiques. Nous proposons d'étudier les approches d'extraction de caractéristiques suivantes :

— ACP (vue en cours).

— Les descripteurs de HOG

Descripteurs de HOG

L'histogramme des gradients orientés (HOG) est un descripteur de forme très répandue dans le domaine de la vision par ordinateur et du traitement d'images [**2**]. Il analyse la distribution des orientations des contours d'un objet pour en décrire la forme et l'apparence. Le gradient d'une image est utilisé pour détecter et caractériser les contours dans une image.
En considérant $f(x, y)$ une image dans un repère orthogonal (Oxy) tel que (Ox) désigne l'axe horizontal et (Oy) l'axe vertical. Le Gradient de l'image en tout pixel de coordonnées (x,y) est donné par :

$$\nabla f = \begin{pmatrix} (\frac{\partial}{\partial x} f \\ \\ \frac{\partial}{\partial y} f \end{pmatrix} \tag{1}$$

Le module du gradient permet de quantifier l'importance du contour, c'est-à-dire l'amplitude du saut d'intensité dans l'image. La direction du gradient est orthogonale à celle du contour. Ainsi, cela permet de déterminer l'arête présente dans l'image. En pratique, le calcul de l'amplitude et de l'orientation du gradient est réalisé en filtrant l'image avec des noyaux comme Sobel et Prewitt.
La méthode HOG consiste à calculer l'amplitude et l'orientation (ou direction) du gradient pour chaque pixel d'une image et de subdiviser l'image en de petites régions. Enfin, le HOG génère un histogramme pour chacune de ces régions séparément.

Pour plus d'informations, vous pouvez consulter le lien suivant : https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/ Pour les approches machine Learning permettant de reconnaitre le visage, nous proposons d'utiliser les arbres de décision et les forêts aléatoires.

3.2 Approches CNN

Les approches CNN permettent de rempalcer la phase d'extraction de caractéristiques et de considérer directement les données images. L'objet de cette section est de concevoir une architecture CNN permettant de reconnaitre un visage.

3.3 Travail demandé

1. Ecrire un programme Python permettant de reconnaitre un visage en utilisant l'ACP.
2. Ecrire un programme Python permettant de calculer les caractéristiques de HOG d'une image.
3. Ecrire un programme Python permettant de reconnaitre un visage en utilisant les descripteurs de HOG et les classifieurs arbre de décision et RandomForest. Ainsi, on distingue deux méthodes : HOG + Decision Tree et HOG + Random Forest.
4. Ecrire un programme Python permettant de reconnaitre un visage en utilisant une architecture CNN.
5. Evaluer les performances de chaque méthode sur un dataset de détection de visage public. Vous pouvez utiliser le dataset du TP sur les eignefaces. Scikit-learn Face dataset. Il faudra fournir un tableau résumant les résultats obtenus (métriques d'évaluation) sur un dataset donné pour chaque méthode de reconnaissance étudiée (méthode utilisant l'ACP, HOG + Decision Tree, HOG + Random forest, Approche CNN)
6. Interpréter les résultats obtenus.
7. **Bonus** : Système complet de reconnaissance de visage qui intègre la phase de détection.

Remarque : Il faudra expliquer le choix des meilleurs hyperparamètres de chaque méthode étudiée.

4 Organisation et rendus

4.1 Langage de programmation et choix des bibliothèques

— **Langage de programmation :** Python
— **Bibliothèques**
  — Pandas, Matplotlib,Seaborn
  — Scikit-learn
  — OpenCV et/ou sckit-image

4.2 Livrables

Le travail sera réalisé en **groupe** de maximum **trois étudiants.** Chaque groupe doit rendre :

— Des notebooks Jupyter (vous pouvez utiliser Goggle Colab Notebook - Google Colaboratory Notebooks) permettant de voir et suivre tout le travail (il doit comporter une explication des méthodes ML non vues en cours, les résultats expérimentaux et leurs interprétations).
— **Ou bien** une archive des scripts pythons et un rapport succinct qui décrit votre projet (il doit comporter une explication des méthodes ML non vues en cours, les résultats expérimentaux et leurs interprétations).
— une présentation pptx de 10 minutes

4.3 Echéances

1. Dépôt du projet sur moodle Version : 01/07/2024 à minuit.
2. Soutenance : Au cours de la dernière séance.

4.4 Barème indicatif

Votre projet sera noté par votre prof pendant la dernière séance du cours en se basant sur votre présentation et le travail fournit.

1. Rapport (clarté, interprétations) : 30%
2. Code (répond aux besoins et qualité) : 35%
3. Soutenance et réponse aux questions : 35%

Remarque. Les membres d'une même équipe peuvent ne pas avoir la même note et cela en fonction de leur degré d'investissement dans le projet. Une petite note contenant la description du degré d'investissement de chaque membre de l'équipe sous forme de pourcentage doit être communiquée à l'enseignant le jour de la soutenance.

Références

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001, pp. I-I, doi : 10.1109/CVPR.2001.990517.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi : 10.1109/CVPR.2005.177.