

Images classification utilisant HOG, le descripteur SIFT et moments de hu

Bakour Imene, Bouchali Nesma Hadia. *Master 2 informatique visuelle, université de science et technologies Houari Boumediene, Alger*

Un des objectifs de la classification d'objets dans des séquences vidéo est de concevoir des systèmes intelligents capables non seulement de capturer des vidéos mais aussi de les interpréter en termes d'objets, de classes et de comportements.

L'objectif de la réalisation de se projet et de concevoir un classifieur d'images basé sur la classification des features extraits des images.

La classification selon « IJuan-Manuel et TORRES-MORENO » dans leurs introductions à la classification des formes, est l'attribution d'une classe spécifique a un objet donne. Cette attribution a besoin d'un certain degré d'abstraction pour pouvoir extraire des généralité s à partir des exemples dont on dispose.

Pour une machine, la classification de donne es est difficile, c'est pour cela que ces dernières années, plusieurs nouvelles techniques d'apprentissage ont été développées. Dans notre travail on utilisera l'une des solutions les plus adopté aujourd'hui : les réseaux de neurones.

Un réseau de neurones artificiels, ou réseau neuronal artificiel, est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques.

I. INTRODUCTION

Les étapes essentielles de réalisation du classifieur peut se résumer en quelques points clés:

- Prétraitement et nettoyage des données
- Extraction des points d'intérêts et des descripteurs
- Intégration des données
- Réalisation du classifieur

II. PRESENTATION DES DONNEE UTILISEES

« Le prétraitement et le nettoyage des données sont des tâches importantes qui doivent intervenir avant d'utiliser un jeu de données à des fins d'apprentissage automatique. » [1]

Pour pouvoir utiliser le jeu de données, on a utiliser le fichier explicatif qui présente les données de chaque classe, pour deviser les données selon les classes. Dans chacune des 9 classes on a des images en RGB et autre Binaires.

Pour garder une équité entre les données des classes, certaines données on était réduites et d'autres augmentés.

III. FEATURES EXTRACTION

A. Moments de Hue

Les moments Hu sont un ensemble de 7 nombres calculés à l'aide de moments centraux invariants aux transformations d'image. Les 6 premiers moments se sont révélés invariants à la translation, à la mise à l'échelle, à la rotation et à la réflexion. Tandis que le signe du 7ème moment change pour la réflexion de l'image.

Les 7 moments sont calculés à l'aide des formules suivantes:

$$\begin{aligned} h_0 &= \eta_{20} + \eta_{02} \\ h_1 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ h_2 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ h_3 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ h_4 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ h_5 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ h_6 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned}$$

B. Descripteur SIFT

Le scale-invariant feature transformé (SIFT) est un algorithme utilisé pour détecter et décrire les caractéristiques locales dans les images numériques. Il localise certains points clés puis leur fournit des informations quantitatives (appelées descripteurs) qui peuvent par exemple être utilisées pour la reconnaissance d'objets. Les descripteurs sont censés être invariants vis-à-vis de diverses transformations sur l'images.

En général, l'algorithme SIFT peut être décomposé en quatre étapes :

- Détection de point caractéristique (keypoint)
- Localisation de points de fonction
- Mission d'orientation
- Génération de descripteurs de fonctionnalités.

C. Egalisation de nombre des features

Le nombre et la position des entités sélectionnés par le détecteur de points SIFT sont différents, donc ceux-ci les fonctionnalités sont spécifiques à la classe. Afin de ne comparer que les paires de caractéristiques avec une signification physique similaire, le même nombre de points d'intérêt construit dans chaque image pour calculer la similitude entre chaque paire est nécessaire comme entrée dans un reseau de neuron. Par consequent, on s'est inspirés d'un travail de reconnaissance facials, dont ils ont proposé d'établir un schéma de regroupement des K-construire les sous-régions automatiquement sur la base de l'emplacements des fonctions dans les échantillons d'apprentissage. Le

[1] Tache de pre paration de donne es pour l'apprentissage automatique, Microsoft Azure, disponible sur: <https://docs.microsoft.com/fr-fr/azure/machine-learning/team-data-science-process/prepare-data>

clustering

Le schéma est le suivant :

(1) Pour la saisie d'images enregistrées, initialisez la sous-région k-centres de cluster avec des valeurs aléatoires.

(2) Décidez de la sous-région la plus proche pour chaque point caractéristique chaque image en utilisant la distance euclidienne et mettre à jour la valeur de chaque centre pour reconstruire les sous-régions.

(3) Si les nouveaux centres restent les mêmes qu'au paravent

recalculé, arrêter le clustering et les k sous-régions restantes sont les zones résultantes pour l'appariement.

(4) Après avoir construit les sous-régions sur l'image du visage, lorsque test d'une nouvelle image, toutes les fonctionnalités SIFT extraites de l'image sont attribués dans les sous-régions correspondantes en fonction des emplacements. [3]

IV. REPRESENTATION DES DONNEES

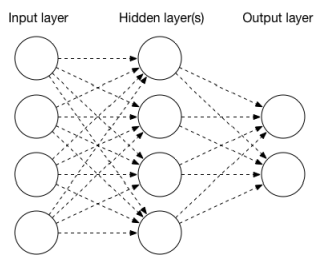
Chaque image est représentée par 135 différentes données dont 128 représente le descripteur SIFT et 7 moments de Hu.

TABLE I
Représentation des données

Id image	Classe	Descripteur SIFT	Moments de Hu
----------	--------	------------------	---------------

V. CONCEPTION DU CLASSIFIEUR RNN

FIGURE I
Exemple d'architecture de réseau de neurone



De nos connaissances, l'architecture de réseau de neurones est de finie par rapport à notre problème d'apprentissage, et l'utilisation de plusieurs couches de neurones d'où l'augmentation de complexité de notre fonction de classification est souvent non nécessaire.

Selon [2], les inputs dans la première couche représentent nos données selon les différentes factures. Ou chaque neurone initial reçoit des entrées.

La couche (i+1) du réseau, se compose généralement de 70% à 120% du nombre de neurones dans la couche avant (i).

Les neurones sur la dernière couche qui est la couche de sorties, représente les sorties possibles que le réseau pourra générer.

A. Solution proposée

- Nos données sont déjà étiquetées donc notre classification sera supervisée
- Transformation de noms de classes en entier id de classe
- Utilisation de 135 neurones dans la couche initiale
- Utilisation de 1 couche cachée au début
- 9 neurones dans la couche output

FIGURE 2
Représentation des changements de valeurs d'exactitudes par epochs

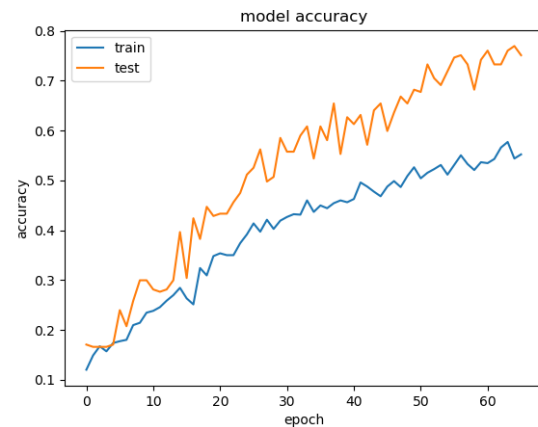
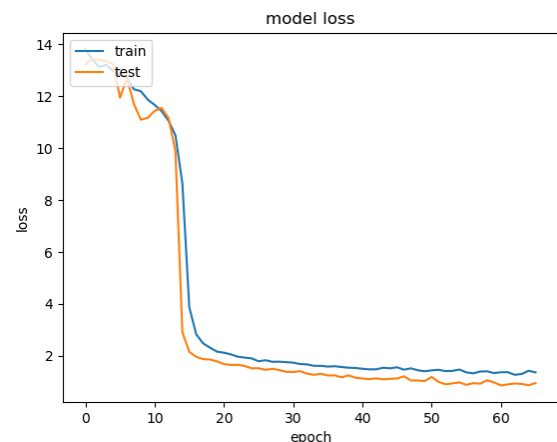


FIGURE 3
Représentation des changements de valeurs de pertes par epochs



De ces graphes on peut remarquer que le classifieur RNN nous a pas donné de bons résultats.

Propositions :

Ce problème peut être dû à cause de :

- Ressemblances entre quelques classes
 - Nombre d'instances d'apprentissage non suffisant
- ce problème nous a encourager à tester d'autres classifieurs.

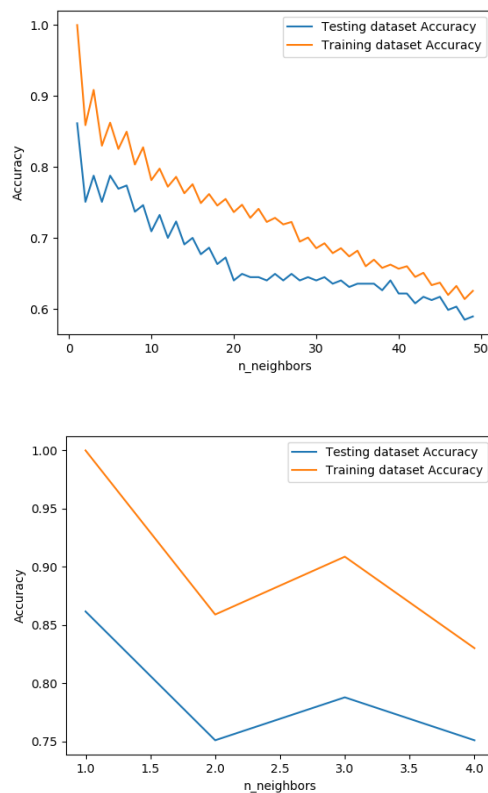
VI. CONCEPTION DU CLASSIFIEUR KNN

L'algorithme K-NN est utilisé pour résoudre les problèmes de classification de modèles K- plus proche voisin ou crée une frontière imaginaire pour classer les données. Lorsque de nouveaux points de données arrivent, l'algorithme essaie de prédire cela au plus près de la ligne de démarcation.

Par conséquent, une valeur k plus élevée signifie des courbes de séparation plus lisses, ce qui donne des modèles moins complexes. Par contre, une valeur k plus petite tend à sur ajuster les données et à produire des modèles complexes. On a testé plusieurs valeurs de k pour trouver celle qui donne de meilleurs résultats.

FIGURE 4 ET 5

Représentations des résultats de classification knn avec différentes valeurs de k



Le bon choix sera alors de prendre la valeur k=3 dans notre cas.

VII. CONCEPTION DU CLASSIFIEUR SVM

Un support-vecteur-machine (SVM) est un classifieur défini par un hyperplan de séparation. En d'autres termes, étant donné les données d'apprentissage étiquetées (apprentissage supervisé), l'algorithme génère un hyperplan optimal qui catégorise de nouveaux exemples. Dans un espace à deux dimensions, cet hyperplan est une ligne divisant un plan en deux parties où chaque classe se trouvait dans un des côtés

CAPTURE 1

Résultat de la classification utilisant le classifieur SVM

```
Using TensorFlow backend.
Accuracy: 0.880184331797235
Precision: 0.880184331797235
Recall: 0.880184331797235
>>> |
```

Fig. 1. Magnetization as a function of applied field. (Note that "Fig." is abbreviated and there is a period after the figure number followed by two spaces.)

Number footnotes separately with superscripts (References | Insert Footnote). Place the actual footnote at the bottom of the column in which it is cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Check that all figures and tables are numbered correctly. Use arabic numerals for figures and Roman numerals for tables.

VIII. CONCEPTION DU CLASSIFIEUR ARBRE DE DECISION (EN PLUS)

CAPTURE 2

Résultat du classification utilisant un arbre de décision

```
Using TensorFlow backend.
Accuracy of Decision Tree classifier on training set: 1.00
Accuracy of Decision Tree classifier on test set: 0.83
>>> |
```

IX. EVALUATION

Classifieur	Rapidité	Meilleure Accuracy
RNN	Moins rapide	Au alentours de 0.75
KNN	Moyen	0.78
SVM	Rapide	0.88
Arbre de décision	Rapide	1

Pour pouvoir comparer réellement entre les classifieur, l'utilisation des autres descripteur est la meilleure solution, mais dans notre cas on a pas pu les utiliser faute de temps.

X. CONCLUSION

Ce projet s'est avéré très utile pour montrer les capacités d'un système capable de détecter et de classer différents objets, soit en images ou une vidéo statique. Nous avons également pris conscience de tous les différents détecteurs de caractéristiques et extracteurs et dans lesquels circonstances, chacun d'entre eux devrait être utilisé, pour la méthodologie, ce projet nous a bien définie l'importance de choix de classifieurs et de descripteur.

[3] PERSON-SPECIFIC SIFT FEATURES FOR FACE RECOGNITION-University, Shanghai, China