

TP4 - vidéos

Dans ce TP nous travaillerons avec un sous-ensemble de vidéos du corpus **UCF Sports** que vous pouvez télécharger depuis la page Moodle du cours.

Le fichier `videos_samples.txt` contient la liste des vidéos du corpus, ainsi que l'étiquette correspondante à chaque vidéo.

Vidéo - séquence d'images

Nous pouvons traiter une vidéo comme une séquence d'images et, par conséquent, nous pouvons voir la description de la vidéo comme étant constituée par les descripteurs extraits sur les images la composant.

Dans la suite, nous nous intéressons aux descripteurs globaux tels que les histogrammes.

Q1/ Ecrire un script python qui calcule les histogrammes couleur pour chaque image d'une vidéo.

Q2/ Ecrire un script python qui calcule un histogramme moyen pour une vidéo, à partir des histogrammes couleur de chaque image la composant.

Q3/ Si l'on considère l'histogramme moyen comme un descripteur pour une vidéo, mettez en place un protocole de classification sur les vidéos de la base **UCF Sports**. L'évaluation se fera en utilisant un processus de cross validation en 4 folds et un classifieur de votre choix.

Veuillez utiliser les configurations suivantes :

- a) utilisez que les données des classes **Diving-Side / Golf-Swing-Front**
- b) utilisez que les données des classes **Kicking-Front / Golf-Swing-Front**
- c) utilisez toutes les classes

Reportez les résultats obtenus. Commentez-les.

Visualiser les flux optiques (en tant qu'images RGB)

Le flux optique permet de rendre compte du mouvement perçu dans la vidéo.

Afin de s'affranchir de la spécificité de la texture/couleur des objets en mouvement, il est intéressant d'utiliser directement l'information de mouvement. Un objet rouge et un objet vert se déplaçant de la même manière, auront des histogrammes de couleurs différentes mais partageront les mêmes propriétés de mouvement.

En partant de cette observation, nous essayerons d'exploiter l'information du mouvement dans le cadre de la classification d'actions, en nous appuyant cette fois-ci sur l'information de mouvement.

Q4/ Ecrire un script python qui calcule les flux optiques entre chaque deux images successives pour une vidéo.

Vous pouvez utiliser la fonction `calcOpticalFlowFarneback` avec les paramètres ci-dessous:

```
cv2.calcOpticalFlowFarneback(previous_frame, next_frame, None, 0.5, 3, 15, 3, 5, 1.2, 0)
```

N'oubliez pas que vous devez procéder à une conversion des images en niveaux de gris à priori. Vous pouvez utiliser la fonction `cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)` pour cela.

Q5/ Pour visualiser les flux optiques, nous passons par une transformation des valeurs (dx, dy) associées à un pixel à une représentation HSV où : * le canal H à la direction du flux $\text{atan2}(dy, dx)$ et * le canal V correspond à la magnitude du flux (la norme 2 du vecteur (dx, dy)) normée elle-même sur l'intervalle $[0..1]$

```
flow_hsv[...,0] = np.arctan2(flow[...,1], flow[...,0])/np.pi*180. + 180.
flow_hsv[...,1] = 1
flow_hsv[...,2] = np.linalg.norm(flow, axis=2, ord=2)
flow_hsv[...,2] = (flow_hsv[...,2] - np.min(flow_hsv[...,2]))
                  / (np.max(flow_hsv[...,2]) - np.min(flow_hsv[...,2]))
```

Selon que vous traitez l'ensemble des trames de flux d'une vidéo ou une trame de flux à la fois, vous devez créer le tensor `flow_hsv` de manière convenable au préalable :

```
#une trame à la fois
flow_hsv = np.empty((flow.shape[0],flow.shape[1],3), float32)

#les trames d'une video à la fois
flows_hsv = np.empty((flow.shape[0],flow.shape[1],flow.shape[2],3), float32)
#dans ce cas, vous devez aussi remplacer axis=2, par axis=3 lors du calcul des normes
```

L'intérêt de transformer l'ensemble des trames à la fois et d'obtenir une normalisation des flux qui tient compte de l'étendue complète des magnitudes observées sur l'intégralité de la vidéo. Sans cela, les faibles magnitudes présentes sur les trames comportant peu de mouvements seront perçues comme fortes lors de la visualisation.

Nous procédons ensuite à une conversion de HSV vers RGB pour visualiser les flux ainsi obtenus. La conversion en RGB va générer des intensités comprises entre `[0..1]` sur chaque canal.

Ecrivez une fonction qui permet de générer et sauvegarder les flux optiques sous formes d'images RGB pour une vidéo. Lors de la sauvegarde des images RGB pensez à multiplier par 255 (car valeurs comprises entre `'0..1']`) et de convertir en entiers (`int`) les valeurs contenues dans les tenseurs. Vous pouvez utiliser la fonction `np.astype` pour convertir tous les tenseurs en entiers.

Pensez à utiliser le dossier `/local` pour réaliser les sauvegardes de vos données.

Vidéos - flux optiques comme orientations et magnitudes

Nous pouvons aussi exploiter les flux optiques directement sans passer par une représentation RGB de ceux-ci.

Nous considérons chaque point du flux comme un vecteur défini par son orientation et sa magnitude.

L'espace de représentation des orientations est borné et s'étale entre 0° et 359° .

En revanche, les magnitudes peuvent avoir des plages de représentation très larges, de part la vitesse de réalisation des actions, ou de part, les erreurs de mesure.

Afin de pouvoir construire des histogrammes qui traitent de l'orientation et de la magnitude conjointement, nous pouvons nous appuyer sur les valeurs min et max des magnitudes observées au sein du corpus de données.

Q6/ Ecrivez une fonction qui transforme un flux optique (`dx,dy`) dans sa représentation '`orientation, magnitude`') en vous servant des règles suivantes:

```
flow_mo[...,0] = (np.arctan2(flow[...,1], flow[...,0])/np.pi*180. + 180.).astype(int)
flow_mo[...,1] = np.linalg.norm(flow, axis=2, ord=2).astype(int)
```

Nous employons ici une conversion vers des `np.array` avec `dtype=int` afin de faciliter la construction d'histogrammes par la suite.

Q7/ Ecrivez une fonction qui extrait l'ensemble de magnitudes observées sur la première vidéo de chaque classe de mouvement et sauvegardez-les dans un fichier.

Q8/ Ecrivez une fonction qui calcule un histogramme à 32 bins de ces magnitudes et visualisez-le afin d'identifier un seuil raisonnable pour les magnitudes apparaissant très rarement afin d'éliminer autant que possible les outliers.

En construisant un deuxième histogramme sur une plage proche de basse magnitudes (comprise en 0 et 8, par exemple), vous pourriez également essayer d'identifier à partir de quel moment il devient intéressant de considérer les flux optiques comme signifiant un véritable mouvement. Parfois, des nombreux points ne bougent pas dans l'image, mais à cause des erreurs de mesure (ouverture, discontinuités), on leur attribue des magnitudes supérieurs à zéro.

Q9/ Ecrivez une fonction qui filtre un flux optique en mettant à zéro les points dont la magnitude est inférieure au seuil minimal et supérieure au seuil maximal identifiés en **Q8**.

Q10/ Ecrivez une fonction qui calcule un histogramme de flux optique en partant de la représentation (**orientation**, **magnitude**) en ignorant les points ayant une **magnitude==0**.

Pour cela vous devrez fragmenter l'espace (**orientation**, **magnitude**). Si vous souhaitez disposer de **o_bins** pour fragmenter l'orientation et de **m_bins** pour fragmenter la magnitude, vous allez construire un vecteur disposant de **o_bins*m_bins** cases.

L'étendue d'une cellule sera de **o_bin_etendue = 360/o_bins** degrés pour les orientations et de **m_bin_etendue = max_mag/m_bins** pour les magnitudes. Vous pourrez ensuite adapter la fonction **color_histogram** (fournie en *TP2*) pour remplir l'historgramme.

Vous pouvez aussi choisir une implémentation moins efficace en parcourant les points composants le flux optiques et en incrémentant la case correspondante **bin_o*m_bins+bin_m**, où : * **bin_o** correspond au bin d'orientation où le point (**o,m**) devra se trouver (**=o/o_bin_etendue**) et * **bin_m** correspond au bin de magnitude où le point (**o,m**) devra se trouver (**=m/m_bin_etendue**).

Q11/ Calculez un histogramme moyen pour une vidéo en partant des histogramme de flux optique calculés entre deux images successives comme indiqué en **Q10**.

Q12/ Réappliquer le protocole de classification de la **Q3** sur ces nouveaux histogrammes moyens.

Q13/ Reappliquer le protocole de classification de la **Q3** en considérant cette fois-ci uniquement des histogrammes d'orientation (en ignorant les points ayant une **magnitude == 0**)