Computer Vision - TP1

Gaspard JUILLET - No. 2485855
Paul HOAREAU - No. 2483853

1 Présentation des méthodes

1.1 HOG - Histogram of Oriented Gradients

La méthode HOG utilise les gradients d'intensité pour décrire une image, en particulier ses formes. L'image est d'abord normalisée. Ensuite, pour chaque pixel, on compare son intensité à celle de ses voisins verticaux et horizontaux. Ces différences permettent de calculer l'intensité et la direction du gradient. L'image est représentée par un vecteur constitué de l'ensemble de ses histogrammes. Ces histogrammes sont générés en divisant l'image en cellules, chaque cellule produisant un histogramme. La largeur des bandes de l'histogramme (ou bins), et donc sa précision, est déterminée par un pas (par exemple 20°). Chaque histogramme de cellule est rempli avec les gradients de ses pixels. Additionnellement, les cellules de l'image peuvent être regroupées en blocs dont on normalise l'intensité pour améliorer la robustesse aux variations d'intensité.
HOG encode principalement les informations de structure et de forme.

1.2 ResNet - Histogram of Oriented Gradients

ResNet est un modèle profond pré-entrainé à la classification d'images sur le dataset ImageNet. Il est construit avec des couches de convolution 2D et ce sont ses *skip connections* qui ont fait de cette architecture une innovation majeure. Ces dernières permettent de limiter le problème du vannishing gradient, et ont donc permis d'entraîner pour la première fois un modèle aussi profond avec succès.
On attend de ses couches finales qu'elles encodent des informations complexes qui permettent d'identifier des concepts clés dans des images, ces mêmes concepts qui ont permis au modèle de classifier des images avec succès pendant l'entraînement.

1.3 Centroïde le Plus Proche & k Plus Proches Voisins

Une fois les features obtenues pour chacune de nos frames, nous devons classifier chacune d'elles dans une des 5 classes. Pour cela nous essayons deux méthodes différentes.

- La méthode du Centroïde le plus proche se base sur un algorithme de machine learning de classification. Pour chaque label, les individus de références sont représentés par un vecteur moyen construit à partir des vecteurs de référence. Pour classifier une donnée test on la compare à chacun des centroïdes et on lui assigne la classe du centroïde le plus proche.
- k-NN est aussi un algorithme de machine learning de classification sans entraînement. Les individus test sont comparés avec tous les individus de référence. Parmi les k individus de référence les plus proches, le label le plus représenté est attribué à l'individu testé.

La méthode du plus proche centroïde ne conserve que les vecteurs moyens, là où la méthode k-NN utilise tous les individus de référence. De ce fait, la méthode k-NN a une complexité spatiale et temporelle supérieure à celle de la méthode des centroïdes les plus proches. Cependant, c'est bien la présence de plusieurs individus de référence qui fait de cette méthode une méthode plus flexible et plus puissante : là où la méthode basée sur les centroïdes se base sur l'idée qu'une classe peut-être représentée par un vecteur moyen, la méthode k-NN permet de capturer des distributions plus complexes.

2 Hypothèses de performance

2.1 Comparaison des extracteurs

Nous prédisons qu'utiliser ResNet comme encodeur produira les meilleurs résultats. En effet, ses couches de convolution 2D et sa profondeur en font une approche puissante pour capturer la complexité des tâches à classifier. La différence sera certainement prédominante sur les classes qui se ressemblent (ex: Violon vs. Cello; Handstand Pushups vs. Pushups).

2.2 Comparaison des classifieurs

Pour la méthode HOG, les embeddings portent des informations importantes qu'on espère spécifiques à chaque classe, mais aussi des informations inutiles sur le background qui n'a à priori pas de lien avec la classe. Là où la méthode k-NN va comparer trop d'informations inutiles (le bruit du background), la méthode des centroïdes va permettre de moyenner ce bruit et de ne garder que ce qui permet vraiment de définir les classes.
Pour ResNet, on peut s'attendre à quelque chose de similaire, mais plus nuancé. En effet, même si le calcul du centroïde fait perdre beaucoup d'information car les différences entre les différents individus d'une classe se retrouvent gommées, il permet encore une fois de limiter l'influence du bruit. Avec le k-NN, on garde l'information sur chacun des individus de référence, et la comparaison avec tous ces individus permet de prendre en compte les disparités au sein de chaque classe de manière plus fine. En sachant que ResNet est pré-entrainé, on pourrait s'attendre à ce qu'il soit capable de filtrer les informations plus ou moins inutiles liées au background, mais que certaines restent. Ainsi, la stratégie k-NN et la stratégie basée sur les centroïdes pourront produire des résultats plus ou moins similaires, avec une possible prédominance des centroïdes si ResNet n'arrive pas à n'extraire que les informations importantes.
k-NN est la méthode avec la plus grande complexité, à la fois spatiale et temporelle. En effet, avec cette dernière on doit garder en mémoire les embeddings de toutes les vidéos de référence, et pour prédire la classe d'une vidéo on doit comparer son embedding à celui de chacun des individus de référence. Avec la méthode du centroïde en revanche, le nombre de vecteurs gardés en mémoire et comparés à une vidéo de test est égal au nombre de classe, ce qui représente un nombre bien plus petit. Pour les mêmes raisons, le temps de prédiction de la méthode k-NN sera plus long que celui de la méthode basée sur les centroïdes.

3 Hypothèses sur la robustesse

3.1 Changements d'éclairages

ResNet sera probablement plus robuste pour les changements d'éclairage que HOG.
ResNet est certainement robuste au changements d'éclairage car c'est un réseau de neurone profond, qui est entraîné dans le but de généraliser et il saura probablement reconnaître les classes en prenant en compte leurs disparités, entre autres les variations d'éclairage. Cette généralisation dépend néanmoins de la taille du dataset d'entraînement, de sa qualité, et de la diversité de ses individus. Or le dataset sur lequel a été pré-entraîné ResNet (ImageNet +10 000 000 d'images) nous permet de penser qu'il sera plus robuste par rapport aux changements d'éclairages que HOG..
Pour ce qui est de HOG, la normalisation de l'image, suivi d'une normalisation de groupes de cellules (blocks) lui confère une certaine robustesse aux changements de luminosité, cependant cette robustesse est intrinsèquement limitée par la forte dépendance de HOG aux variations locales de luminosité de l'image. Une saturation de certaines portions de l'image (augmentation de la luminosité à des endroits où les pixels sont déjà à 255) ou l'effacement de certaines parties (diminution de la luminosité à des endroits où les pixels sont déjà à 0) peut avoir des effets destructeurs pour cette approche.

3.2 Changements géométriques

Pour les augmentations géométriques, on s'attend aussi à ce que ResNet soit plus robuste.
Ici encore, la robustesse de ResNet dépend de son dataset d'entraînement et des augmentations appliquées durant son entraînement. On peut s'attendre à une robustesse limitée car les augmentations vont faire sortir nos images de leur zone normale de distribution. Exemple: un handstand pushup est par définition vertical,

une rotation de 90° va certainement faciliter la confusion entre cette classe et celle des pushups.
Pour ce qui est de HOG on peut s'attendre à ce que cette approche soit très peu robuste aux changements géométriques. En effet, HOG s'appuie beaucoup sur l'orientation des formes dans l'image. En altérant ces informations, la classification par HOG va certainement échouer dans de nombreux cas.

4 Description des expériences et Implémentation

4.1 Échantillonnage

Pour réaliser nos expériences, nous avons utilisé le sous-ensemble demandé du dataset UCF101 comprenant 5 classes: Biking, HandstandPushups, PlayingCello, PlayingViolin et PushUps.
Pour créer nos ensembles de référence (train) et de test, nous n'avons pas juste mélangé toutes les images. **Nous avons séparé les données en fonction des groupes de vidéos (indiqués par le préfixe 'g' dans les noms de fichiers)** pour éviter qu'une image d'une même vidéo source se retrouve en référence et une autre image de la même vidéo source en test (ce qui correspondrait à un *data leakage*). Nous avons utilisé un ratio de 0.75, c'est-à-dire que 75% des groupes de vidéos servent de référence et 25% pour le test.
Pour extraire les caractéristiques, nous n'avons pas pris toutes les images des vidéos car ce serait trop lourd. Nous avons échantillonné les vidéos en prenant 1 trame par seconde (*frame rate* = 1). Sachant que nous ne cherchons pas à observer la dynamique du mouvement mais seulement des observations statiques des actions, un échantillionnage plus important n'apporterait pas d'information.

4.2 Implémentation

Nous avons utilisé Python avec les bibliothèques : `OpenCV` pour lire les vidéos, `scikit-image` pour le calcul des HOG, `torch` et `torchvision` pour le modèle ResNet, et `scikit-learn` pour les métriques.
Pour la méthode HOG, nous avons utilisé les paramètres suivants : 8 orientations, 16x16 pixels par cellule et 1x1 cellule par bloc.
Pour la méthode ResNet, nous avons utilisé le modèle `resnet50` pré-entraîné (en prenant les poids par défaut). Nous avons enlevé la dernière couche (la couche *fully connected*) pour récupérer le vecteur de caractéristiques (embedding) juste avant la classification.
Afin de comparer les méthodes entre elles, il nous a fallu fixer une valeur de l'hyperparamètre k pour la méthode de K-NN. Pour cela nous avons fait varier K et observé l'accuracy de chacun des deux modèles (HOG et ResNet) en fonction de ce dernier, dans l'optique de sélectionner pour la suite celui aboutissant aux meilleurs résultats.

Les résultats obtenus sont présentés dans les graphes suivants :

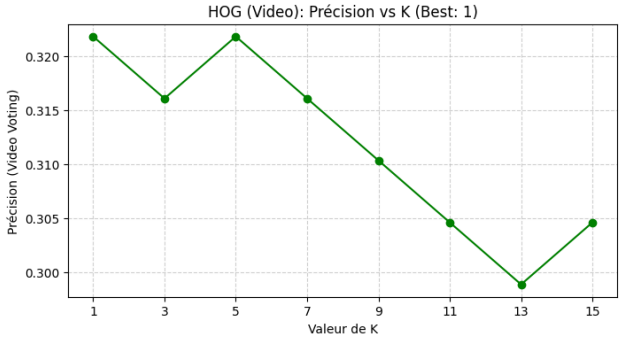

Figure 1: Précision en fonction de k (HOG)

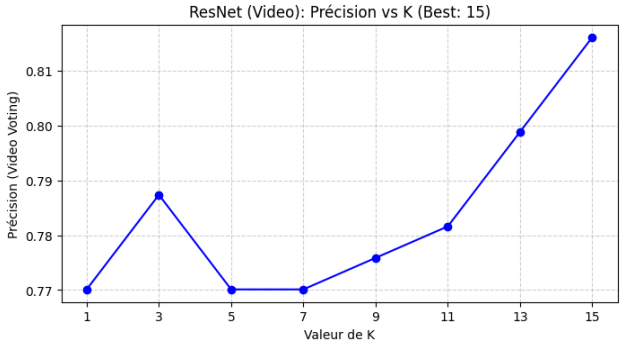

Figure 2: Précision en fonction de k (ResNet)

On observe que pour HOG, on atteint le maximum d'accuracy pour les valeurs $k = 1$ et $k = 5$. Nous avons conservé $k = 5$ pour la suite de nos expériences, considérant que $k = 1$ était moins propice à la généralisation. Pour ResNet, on observe qu'à partir de k = 5, l'accuracy est croissante en fonction de k. Nous avons décidé de conserver la valeur $k = 15$ sans chercher à tester de valeur plus importante, afin de ne pas avoir de complexité trop élevée.
Enfin, pour la classification des vidéos complètes, nous avons implémenté un système de vote majoritaire : on classifie chaque image de la vidéo test, et la classe qui revient le plus souvent est attribuée à la vidéo.
Nous avons fait le choix de ne pas faire de cross-validation lors de nos expériences car nous avons considéré que, compte tenu de l'ensemble des frames constituant chaque vidéo, le dataset avait une taille suffisante dans son ensemble pour que cette étape ne soit pas nécessaire et que l'utilisation classique d'un split train / test aboutisse à des résultats fiables.

4.3 Métriques d'évaluation

Pour évaluer nos modèles, nous utilisons principalement l'accuracy: le ratio entre le nombre de vidéos correctement classées et le nombre total de vidéos.
Nous utilisons aussi la matrice de confusion pour mieux comprendre les erreurs, notamment pour voir quelles classes sont confondues entre elles (par exemple si le modèle confond le violon et le cello). Nous regardons aussi la précision, le rappel et le f1-score pour avoir des détails par classe.

5 Résultats d'Expérimentation

Voici les résultats obtenus sur l'ensemble de test (comportant 167 vidéos au total) en utilisant le vote majoritaire sur les trames.

Méthode d'extraction	**Stratégie de classification**	**Accuracy (Exactitude)**
HOG	Centroïde	0.43
HOG	k-NN ($k = 5$)	0.32
ResNet	Centroïde	0.89
ResNet	k-NN ($k = 15$)	0.82

Table 1: Comparaison des taux d'exactitude par méthode

5.1 Meilleure performance et Matrice de confusion

La combinaison qui fonctionne le mieux est ResNet avec le classifieur par Centroïde, avec une exactitude de 89%.
En analysant les résultats détaillés de cette méthode et la matrice de confusion, on observe que :

- La classe "PlayingCello" est très bien reconnue (F1-score de 0.92). ResNet doit être capable de faire la différence entre un violon et un cello.

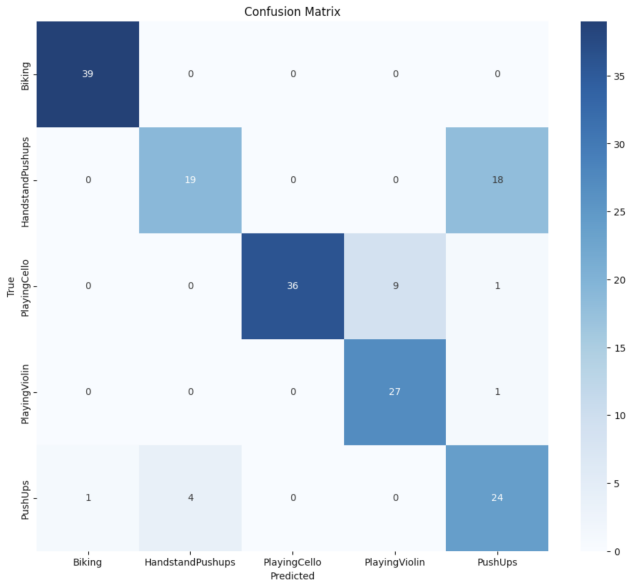

Figure 3: (Seaborn) Matrice de Confusion - ResNet Centroïde

- La classe "Biking" fonctionne bien aussi (F1-score de 0.93).
- La principale confusion se situe au niveau des HandstandPushups et des PushUps. Le modèle a un rappel plus faible pour HandstandPushups (0.62), ce qui signifie qu'il en rate beaucoup, et une précision plus faible pour PushUps (0.65), ce qui veut dire qu'il prédit PushUps alors que le vrai label est HandstandPushups. ResNet a été entraîné pour reconnaître des humains, mais faire la distinction entre les actions performées par ces humains sans l'intervention d'un objet supplémentaire s'avère plus compliqué.

5.2 Résultats de Robustesse

Pour vérifier nos hypothèses de la section 3, nous avons lancé les modèles sur deux variantes du jeu de test : une avec la luminosité modifiée (soustraction de valeur sur le canal V de HSV) et une avec une rotation de 90 degrés.

Voici un tableau récapitulatif des taux d'exactitude (accuracy) obtenus :

Modèle	**Original**	**Luminosité (- 50)**	**Rotation (90°)**
HOG + Centroïde	0.55	0.44	0.31
HOG + k-NN	0.35	0.35	0.20
ResNet + Centroïde	0.81	0.80	0.55
ResNet + k-NN	0.78	0.80	0.61

Table 2: Comparaison de la robustesse des modèles

On constate que la rotation détruit complètement les performances de HOG (puisqu'une accuracy de 0.20 équivaut à du hasard pour 5 classes), tandis que la variation de luminosité à un impact limité. ResNet résiste encore mieux à la luminosité (quasiment aucune perte) mais souffre quand même de la rotation.

6 Discussion et Analyse

6.1 Analyse des performances (Retour sur les hypothèses)

Dans la section 2, nous avions prédit que ResNet serait meilleur que HOG. Les résultats confirment cette hypothèse de manière flagrante (81% contre 55% pour les meilleures configurations). HOG a eu beaucoup de mal à généraliser, probablement parce que les arrière-plans des vidéos (gymnase, extérieur...) variaient trop et que HOG encode tout ce qu'il voit, y compris le décor inutile.

Par contre, nous avions prédit que k-NN pourrait être meilleur que l'approche Centroïde pour ResNet. Nos résultats montrent le contraire (Centroïde à 0.81 vs k-NN à 0.78).
Cela peut s'expliquer par le fait que le vecteur moyen (centroïde) permet de lisser le bruit venant du background. Comme une vidéo contient beaucoup d'images qui se ressemblent, k-NN est peut-être plus sensible aux outliers (images atypiques) alors que le centroïde représente mieux le concept global de l'action sportive.

Pour HOG, comme prévu, le Centroïde est bien meilleur que k-NN (0.55 vs 0.35). Avec k-NN et HOG, le modèle semble se perdre dans la comparaison pixel par pixel (ou cellule par cellule) des bruits de fond, alors que la moyenne par classe aide à faire ressortir la forme humaine en mouvement.

6.2 Analyse de la matrice de confusion

Notre meilleur modèle est formé de la combinaison "ResNet + Centroïde". En regardant ses résultats détaillés :

- **Distinctions faciles :** "Biking" et "PlayingCello" sont très bien reconnus (Rappel proche de 1.00). Ces actions ont des contextes visuels très forts (le violoncelle, le vélo) que ResNet détecte probablement comme des objets spécifiques.
- **Confusions :** La plus grosse erreur se trouve entre "PushUps" et "HandstandPushups". C'est logique car ce sont deux mouvements qui impliquent souvent les mêmes environnements (tapis de sol, salle de sport) et pas d'objet supplémentaire facilement reconnaissable. La seule différence majeure est l'orientation du corps, ce qui est subtil pour un réseau qui est souvent invariant à la translation et qui n'a pas été entraîné pour repéré des différences aussi fines.

6.3 Analyse de la robustesse

Luminosité :
Nos résultats valident l'hypothèse que ResNet est plus robuste. Son score ne bouge presque pas (0.81 $\rightarrow$ 0.80). HOG, lui, chute de 0.55 à 0.44, tout en restant capable de reconnaître certaines classes.
HOG est basé sur les différences d'intensité entre les pixels (gradients). En changeant fortement la luminosité, on peut saturer certaines zones ou effacer des contrastes faibles, ce qui modifie l'histogramme. ResNet, ayant été entraîné sur ImageNet avec des augmentations de données variées, a appris à ignorer l'éclairage pour se concentrer sur les objets.

Rotation :
La rotation de 90 degrés est catastrophique pour HOG (chute à 0.31/0.20). C'est normal : HOG calcule des orientations de gradients (vertical, horizontal, diagonal). Si on tourne l'image, une ligne verticale devient horizontale. L'histogramme change complètement et ne correspond plus au modèle de référence.
ResNet résiste mieux (0.55) mais chute quand même lourdement. Cela montre que les CNN ne sont pas naturellement invariants à la rotation. Un HandstandPushup tourné à 90 degrés ressemble peut-être trop à un Pushup normal, ce qui aggrave la confusion entre ces deux classes.

6.4 Compromis et Recommandations

Temps de calcul: Pour chaque méthode on a les temps de calcul suivant:

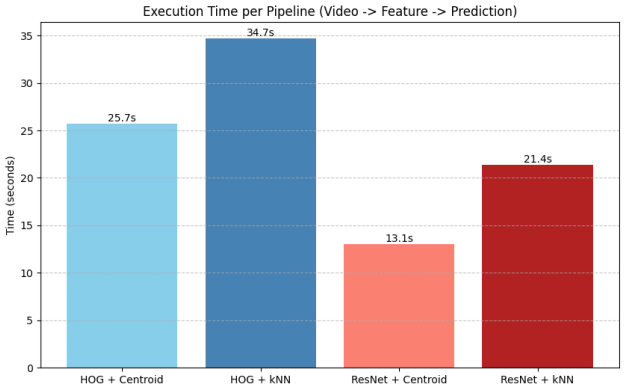

Figure 4: Enter Caption

Si nous devions recommander un système pour une application réelle :

1. **Choix de l'extracteur :** Sans hésitation **ResNet**. Bien que l'extraction soit plus lourde en calcul (nécessite un GPU pour être rapide), la différence de précision est trop grande par rapport à HOG. De plus, la robustesse aux changements de lumière est critique pour une application réelle.
2. **Choix du classifieur :** Nous recommandons la méthode du **Centroïde**.
   - Elle est plus précise dans nos tests.
   - Elle est beaucoup plus rapide en prédiction (une seule comparaison par classe au lieu de comparer avec toutes les images de la base de données comme k-NN).
   - Elle prend très peu de mémoire (on ne stocke qu'un vecteur par classe).

La combinaison **ResNet + Centroïde** offre donc le meilleur compromis entre exactitude, robustesse et vitesse d'exécution au moment du test.