

INF6804 Vision par ordinateur

H2024 – Travail Pratique #1

Description et comparaison de régions d'intérêt

Objectifs:

- Se familiariser à la manipulation de données et au traitement d'images à l'aide de bibliothèques logicielles spécialisées pour la vision par ordinateur.
- Permettre à l'étudiant de se familiariser avec la description d'images basée sur les Matrices de Co-occurrence (MC).
- Permettre à l'étudiant de se familiariser avec la description d'images basée sur les Modèles Binaires Locaux (LBP).
- Apprendre à comparer des vecteurs de caractéristiques.
- Expliquer les avantages et les inconvénients des deux approches de description d'images.

Remise:

- *Tout votre code source* (nous devrions pouvoir exécuter vos tests)
- Un rapport (*.pdf format* de 8 à 12 pages avec une taille de police de 10)
- Remise avant le 9 février 2024, 17h00, sur Moodle –*aucun retard accepté*
- *Vous devez aussi soumettre votre rapport sur TurnItIn!*
 - Enregistrez-vous sur www.turnitin.com en utilisant les informations sur Moodle!

Références:

- Voir les notes de cours sur Moodle (Chapitre 2)

Autres directives:

- Les TP doivent obligatoirement être faits en équipe de deux, ne remettez qu'une seule version du rapport/code!
-

Présentation

Dans ce TP, vous devez caractériser deux méthodes de description de régions d'intérêt dans des images et déterminer quelle méthode est la meilleure et dans quelles circonstances. Une description de votre travail, de vos expériences, et les réponses aux questions de cet énoncé doivent se retrouver dans un rapport.

Vous devez comparer deux méthodes de description: La première méthode se base sur **les Matrices de Co-occurrence (MC)**, la seconde est basée sur **les Modèles Binaires Locaux (LBP)**. Vous pouvez utiliser vos notes de cours comme référence pour comprendre leur fonctionnement de base. Pour plus de détails, allez creuser sur internet! Pour comparer les deux méthodes, vous devrez résoudre une problématique de **recherche d'image par le contenu (CBIR)**. Vous devez caractériser chaque image avec les descripteurs dérivés de MC et LBP et ensuite trouver les images qui correspondent à chaque image requête en mesurant la similarité entre ces descripteurs. On vous demande de comparer les deux méthodes en termes d'efficacité (temps de calcul) et en termes de précision. Vous devez évaluer leurs performances en vous basant uniquement sur le résultat des **3 images les plus similaires** qui ont été extraites de la base de données. Vous êtes libres de choisir les métriques qui, selon vous, correspondent le mieux aux méthodes à implémenter.

En utilisant les données fournies dans l'archive "**data.zip**", vous devez inclure les éléments suivants dans votre rapport (noté sur 20 pts):

1. Présentation des deux approches à comparer (3 pts):
Dans vos propres mots, donnez la description générale et les principes de vos deux approches.
2. Hypothèses de performance pour des cas spécifiques (2 pts):
Identifiez, selon votre compréhension théorique des deux approches étudiées uniquement, quelle approche devrait être meilleure que l'autre en fonction d'au moins DEUX cas d'utilisation. Par exemple, quelle est la meilleure approche à utiliser si le contenu des régions comparées est relativement uniforme? Pourquoi?
3. Hypothèses de performance concernant les boîtes englobantes (bounding boxes) (1 pts):
En sachant que vous avez accès à la boîte englobant l'objet d'intérêt dans chaque image, attendez-vous une amélioration de la performance pour les deux méthodes? Pourquoi?
4. Description des expériences, des données et critères d'évaluation (3 pts):
Décrivez en détail les expériences réalisées pour vérifier les hypothèses des points 2 et 3. Quelles sont les difficultés des images requêtes de la base

de données? Quelles mesures de similarité entre descripteurs avez-vous choisi? Quelle est votre métrique d'évaluation?

5. Description des deux implémentations utilisées (2 pts):

Décrivez l'implémentation des deux approches étudiées. Si vous n'avez pas tout écrit le code vous-même, d'où provient-il? A-t-il demandé des modifications? Sinon, de quels articles (ou sites web) vous êtes-vous inspiré pour l'écrire? Dans tous les cas, quels sont les paramètres principaux utilisés? Comment ont-ils été choisis?

Effectuez une recherche de paramètres optimaux pour affiner 'P' (nombre de points d'échantillonnage) et 'R' (rayon) pour LBP, 'niveaux' (quantification des niveaux) et choix des canaux pour MC, afin de garantir une performance optimale des descripteurs.

Vous pouvez définir les boîtes englobantes manuellement ou par calcul.

6. Présentation des résultats de tests (3 pts):

Donner les résultats d'évaluation tirés de vos expériences en lien avec les hypothèses du deuxième et troisième point. Utilisez un format approprié — tableaux, figures, ...

7. Discussion des résultats et retour sur les hypothèses (3 pts):

Discutez des résultats du sixième point en fonction des hypothèses du deuxième et troisième point. Quelles hypothèses sont supportées par les résultats? Lesquelles ne le sont pas? Quels tests sont restés sans conclusions? Quels tests pourriez-vous améliorer, et comment?

Évaluez également comment les méthodes MC et LBP ont performé sur '*dolphin_query.png*' et considérez si un autre descripteur abordé en classe pourrait être plus efficace pour ce cas particulier.

8. Lisibilité et complétude (3 pts):

À part le contenu, le format doit être soigné et complet.

Lors des séances de laboratoire, n'hésitez pas à poser des questions — on peut vous aider avec tout problème technique si vous travaillez sur Windows/Linux, ou bien avec votre code si vous travaillez avec du C/C++ ou avec Python/Matlab.

Vous serez **pénalisés de 50% de la note totale si vous ne remettez pas votre code**. De plus, si votre rapport n'est pas remis sur **Turnitin**, celui-ci ne sera pas corrigé. L'ordre de la présentation n'est pas important, tant que tous les éléments ci-haut sont présents. N'oubliez pas de citer vos références!

Ressources

Librairies de vision par ordinateur:

- OpenCV (https://docs.opencv.org/4.0.0/d9/df8/tutorial_root.html)
- scikit-image (https://scikit-image.org/docs/stable/auto_examples/index.html)

Python:

- Guide (<https://wiki.python.org/moin/BeginnersGuide/Programmers>)
- NumPy (<https://docs.scipy.org/doc/numpy/user/quickstart.html>)

Matlab:

- Guide (http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf)
- Aide-mémoire (<http://web.mit.edu/18.06/www/Spring09/matlab-cheatsheet.pdf>)