



UNIVERSITÉ DE MONTPELLIER

MASTER 1 IMAGINE

Projet Image et Compression

Compte Rendu N°6

FONTAINE Emmanuel
HARCHA Ibrahim

7 avril 2024

Introduction

Pendant cette sixième semaine, nous avons exploré les méthodes d'apprentissage automatique pour détecter la falsification d'images. Nous avons opté pour l'utilisation d'un CNN (Convolutional Neural Network), une approche réputée pour son efficacité à extraire des caractéristiques pertinentes des images, ce qui le rend particulièrement adapté pour la détection de falsifications.

1 Trouver une bonne base de données

Pour entraîner notre modèle de détection de falsification, nous devons trouver une base de données appropriée contenant à la fois des images authentiques et falsifiées.

Par ailleurs, chaque image dans la base de données doit être accompagnée d'une étiquette qui indique si l'image est authentique ou falsifiée. Cette carte de vérité terrain est essentielle pour l'entraînement supervisé du modèle, car elle fournit les annotations nécessaires pour apprendre à distinguer les images authentiques des images falsifiées.

En plus de l'ensemble d'entraînement, nous allons devoir avoir des ensembles de validation et de test distincts pour évaluer les performances du modèle. Ces ensembles doivent être séparés de l'ensemble d'entraînement pour évaluer la capacité du modèle à généraliser à de nouvelles données.

2 Prétraitement des données [data_preprocessing.py]

Avant d'entraîner notre modèle, nous allons devoir effectuer plusieurs étapes de prétraitement sur nos données. Cela peut inclure le redimensionnement des images, la normalisation des pixels, et d'autres techniques pour améliorer la qualité et la consistance des données.

3 Construction du modèle CNN [cnn_construction.py]

Nous avons conçu notre modèle CNN en utilisant TensorFlow. Le CNN comprend des couches de convolution, des couches de pooling et des couches entièrement connectées pour extraire et classer les caractéristiques des images.

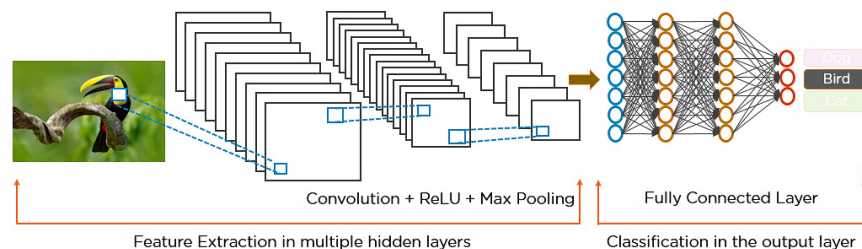


FIGURE 1 – CNN opérations.

Les couches de convolution extraient des caractéristiques significatives des images en utilisant des filtres pour détecter des motifs tels que les bords et les textures. Ces caractéristiques sont essentielles pour la classification et la détection d'objets.

Les couches de pooling réduisent la taille des représentations des caractéristiques, ce qui va permettre de contrôler le nombre de paramètres du modèle et d'améliorer sa vitesse de calcul, tout en réduisant le risque de surajustement.

Les couches entièrement connectées réalisent la classification finale en combinant les caractéristiques extraites, jouant ainsi un rôle crucial dans la prédiction des classes ou des probabilités pour chaque classe possible. En somme, ces couches travaillent ensemble pour permettre au modèle CNN d'apprendre des représentations hiérarchiques des images et de réaliser des tâches de vision par ordinateur avec précision.

4 Entraînement du modèle [model_training.py]

Nous allons entraîner notre modèle CNN sur les données prétraitées en utilisant des techniques d'optimisation, nous allons aussi surveiller les métriques de performance telles que la perte et l'exactitude pour évaluer la capacité de notre modèle à généraliser aux données non vues.

Conclusion

En conclusion, au cours de cette sixième semaine, notre attention s'est entièrement concentrée sur la mise en œuvre et la compréhension du fonctionnement d'un CNN (Convolutional Neural Network) pour notre projet de détection de falsifications d'images. Nous avons consacré notre temps à explorer les mécanismes sous-jacents de cette architecture de réseau de neurones, ainsi qu'à apprendre comment l'adapter efficacement à notre problème spécifique. Notre objectif principal était de comprendre le processus d'entraînement d'un CNN, y compris la manière dont il extrait automatiquement des caractéristiques pertinentes des images et comment il est utilisé pour classer les données.

Sources

<https://datascientest.com/convolutional-neural-network>
<https://www.youtube.com/watch?v=QzY57FaENXg>