

Rapport de méthodologie

Reconnaissance faciale pour contrôle d'accès

via YOLO et CNN à deux couches de traitement

Réalisé par :

Nabil MICHRAF
Mohamed EL MRABET
Ismail AARABI
Hamza BOUKHERYS

Encadré par :

Mr. Hicham JANATI

1. Problématique

La reconnaissance faciale est devenue un élément central des systèmes de sécurité modernes, permettant un contrôle d'accès automatisé et sans contact. Cependant, les environnements réels présentent de nombreux défis : variations de luminosité, angles de prise de vue, expressions faciales, et interférences visuelles (masques, lunettes, ombres). Ces facteurs peuvent altérer la précision des modèles et engendrer des faux positifs ou négatifs.

L'objectif de ce projet est de concevoir une **architecture modulaire** de reconnaissance faciale reposant sur deux couches distinctes :

- Une première couche de détection basée sur le modèle **YOLO (You Only Look Once)**, pour localiser rapidement les visages dans une image ou une vidéo en temps réel ;
- Une deuxième couche de classification **CNN (Convolutional Neural Network)**, pour déterminer si le visage détecté correspond à une personne autorisée à accéder au système.

Cette approche vise à combiner la vitesse de YOLO avec la robustesse discriminante du CNN, garantissant une détection rapide, précise et sécurisée.

2. Objectifs du projet

- Concevoir un pipeline complet de reconnaissance faciale combinant détection et classification.
- Exploiter un modèle **YOLOv5 ou YOLOv8 pré-entraîné** pour la détection rapide des visages.
- Entraîner un **CNN supervisé** sur un dataset de visages labellisés (« autorisé » / « non autorisé »).
- Intégrer une phase d'inférence optimisée pour un usage embarqué (GPU léger ou CPU).
- Évaluer la performance selon des métriques robustes (précision, rappel, F1-score, AUC-ROC).
- Étudier l'impact de paramètres clés : taille d'image, seuil de détection, profondeur du CNN, et stratégies d'augmentation.

3. Données utilisées

Le projet s'appuie sur des images faciales issues de bases publiques telles que **LFW (Labeled Faces in the Wild)**, **CelebA**, ou un dataset interne d'accès autorisé/non autorisé.

Chaque image est associée à une étiquette binaire :

- 1 → visage autorisé (accès permis)
- 0 → visage non autorisé (accès refusé)

Les données seront prétraitées pour assurer une cohérence de taille, de luminosité et de cadrage. Une séparation stricte entre entraînement, validation et test sera appliquée pour éviter les fuites de données entre identités.

4. Méthodologie prévue

4.1 Prétraitement et augmentation

- Détection initiale des visages via YOLO, pour extraire uniquement les zones pertinentes.
- Redimensionnement des images à 128×128 ou 224×224 pixels selon le CNN choisi.
- Normalisation des valeurs de pixels (0–1 ou 1–1).
- Augmentation : flip horizontal, rotations 10° , variations de luminosité/contraste, flou léger, occlusions partielles.

Les transformations ne sont appliquées que sur le jeu d’entraînement.

4.2 Couche 1 : Détection par YOLO

- Modèle : **YOLOv5** ou **YOLOv8** pré-entraîné sur **WIDER FACE**.
- Objectif : localiser et découper les visages dans chaque image ou flux vidéo.
- Sorties : coordonnées des bounding boxes et score de confiance.
- Paramétrage : seuil de confiance = 0,5 ; suppression NMS pour éliminer les doublons.

Cette étape constitue le pré-filtrage spatial du pipeline.

4.3 Couche 2 : Classification CNN

- Entrée : visage détecté par YOLO.
- Architecture : Convolutions + BatchNorm + ReLU, Pooling, Dropout, Fully Connected, Sigmoid.
- Optimisation : Adam, taux d’apprentissage = $1e-4$, perte binaire cross-entropy.
- Entraînement : 50 époques maximum avec *early stopping*.

Une version fine-tuned à partir de **VGG16** ou **ResNet18** sera testée pour améliorer la généralisation.

4.4 Étude d’ablation

- A1 — Taille d’entrée : 128×128 vs 224×224
- A2 — Backbone CNN : modèle simple vs VGG16 vs ResNet18
- A3 — Data augmentation : sans / avec occlusions / rotation

- A4 — Seuil de décision : fixe (0,5) vs optimisé sur la courbe Precision-Recall

4.5 Optimisation pour l'inférence

- Conversion du modèle CNN en format **ONNX** puis quantization **FP16** / **INT8** (ONNX Runtime ou TensorRT).
- **Batching** et **fusion de couches** pour accélérer le traitement en temps réel.
- Cible : GPU T4 (Colab) ou Jetson Nano pour déploiement embarqué.

4.6 Évaluation et validation

Métriques principales :

- Accuracy, précision, rappel, F1-score, AUC-ROC.
- Matrice de confusion et courbe Precision-Recall.
- Visualisation **Grad-CAM** pour valider les zones activées sur le visage.

Une analyse qualitative sera menée sur les cas d'erreur : mauvais cadrage, faible éclairage, visages masqués.

5. Conclusion

Ce projet propose une approche modulaire et temps réel de reconnaissance faciale combinant YOLO pour la détection et CNN pour la classification binaire.

La séparation claire entre la localisation et la décision permet :

- une réduction des faux positifs ;
- une montée en échelle simple (ajout de nouvelles classes autorisées) ;
- une adaptation à des environnements contraints.

Les optimisations d'inférence et la quantization assurent la compatibilité avec des déploiements embarqués ou à faible puissance.

6. Bibliographie

- Redmon, J. et al. (2016). *You Only Look Once : Unified, Real-Time Object Detection*. CVPR.
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4 : Optimal Speed and Accuracy of Object Detection*. arXiv.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition (ResNet)*. CVPR.
- Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG)*. ICLR.
- Parkhi, O. M., et al. (2015). *Deep Face Recognition*. BMVC.
- ONNX Runtime. Quantization. <https://onnxruntime.ai/>

— NVIDIA TensorRT. Developer Guide. <https://developer.nvidia.com/tensorrt>