IoT Door Guard

Mini-Projet de Contrôle d'Accès Intelligent

Projet de Développement Technologique

Trabelsi Dhia Eddine
Étudiant en Informatique

19 décembre 2024

Résumé

Ce mini-projet présente un système de contrôle d'accès innovant basé sur la reconnaissance faciale utilisant l'Internet des Objets (IoT). Le système combine un module de reconnaissance faciale avec des technologies de deep learning, créant une solution de sécurité intelligente et automatisée pour le contrôle d'accès. Développé avec Angular CLI, il comprend des tests unitaires et end-to-end pour assurer la fiabilité.

Mots-clés : IoT, Reconnaissance Faciale, Deep Learning, Sécurité, Angular CLI

1 Introduction

1.1 Contexte

L'objectif de ce projet est de développer un système de contrôle d'accès intelligent qui utilise la reconnaissance faciale pour identifier et autoriser l'entrée des personnes autorisées. En utilisant des technologies modernes comme le deep learning et l'IoT, ainsi qu'Angular CLI pour le développement, nous créons une solution de sécurité plus efficace et conviviale que les méthodes traditionnelles.

1.2 Objectifs du Projet

Les principaux objectifs sont :
— Créer un système de reconnaissance faciale fonctionnel
— Implémenter un mécanisme de contrôle d'accès automatique
— Démontrer l'intégration des technologies IoT et de deep learning
— Développer une solution de sécurité pratique et innovante

2 Architecture du Système

2.1 Composants Principaux

Le système est composé de trois composants principaux :
— Module de Capture et de Reconnaissance (Python)
— Serveur de Gestion (Node.js)
— Interface Front-End (Angular)

2.2 Flux de Travail

1. Capture d'image via la caméra
2. Détection et prétraitement du visage
3. Comparaison avec les visages enregistrés
4. Décision d'accès (autorisé/refusé)
5. Enregistrement de l'événement

3 Implémentation Technique

3.1 Reconnaissance Faciale

Exemple de code de base pour la reconnaissance faciale :

```
import cv2
import face_recognition

def recognize_face(known_faces, unknown_face):
    matches = face_recognition.compare_faces(known_faces,
       unknown_face)

    if True in matches:
        return "Acc s Autoris "
    else:
        return "Acc s Refus "

def access_control():
    camera = cv2.VideoCapture(0)
    known_faces = load_known_faces()

    while True:
        ret, frame = camera.read()
        unknown_face = face_recognition.face_encodings(frame)[0]

        result = recognize_face(known_faces, unknown_face)
        print(result)
```

Listing 1 – Reconnaissance Faciale Simple

3.2 Communication IoT

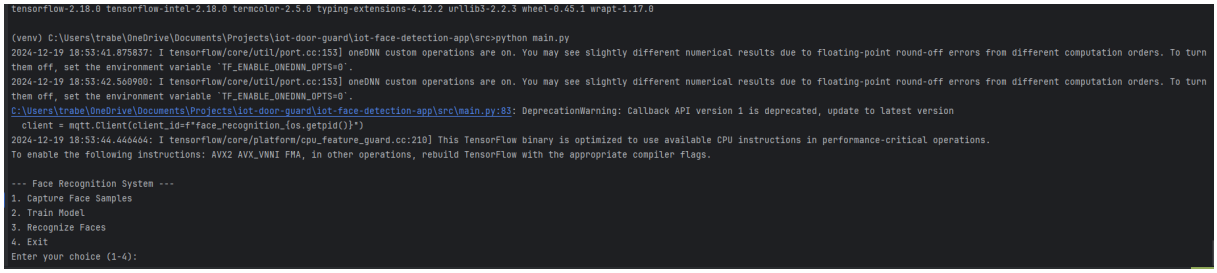
Pour la communication IoT, nous utilisons MQTT :

```
import paho.mqtt.client as mqtt

def on_connect(client, userdata, flags, rc):
    print("Connect  au broker MQTT")

def send_access_event(status):
    client = mqtt.Client()
    client.on_connect = on_connect
    client.connect("broker.hivemq.com", 1883)
    client.publish("access/control", status)
```

Listing 2 – Communication MQTT Simple

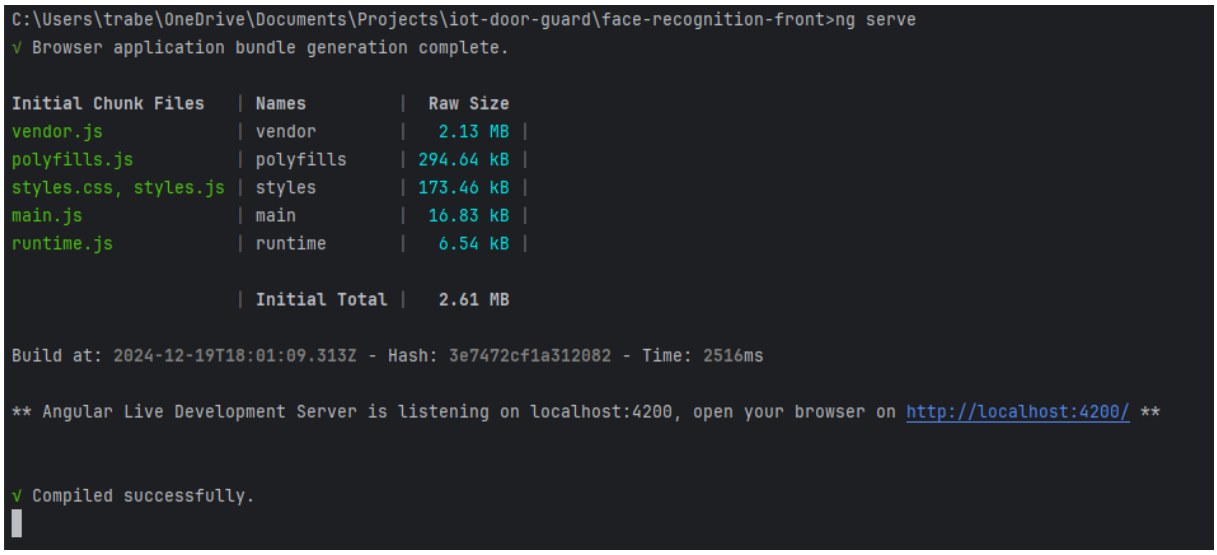
4 Démonstration du Système

4.1 Interface Terminal



(a) Terminal de l'application IoT

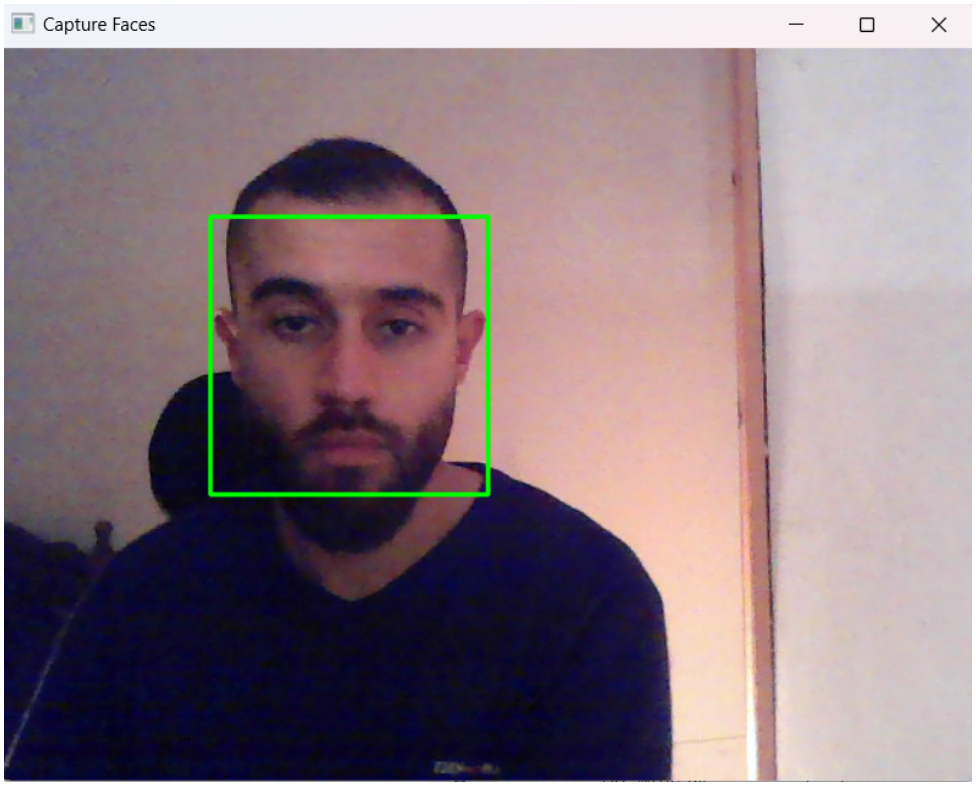
(b) Terminal du serveur backend



(c) Terminal du serveur frontend

FIGURE 1 – Interfaces terminales de l'application

4.2 Processus d'Ajout d'un Nouveau Visage



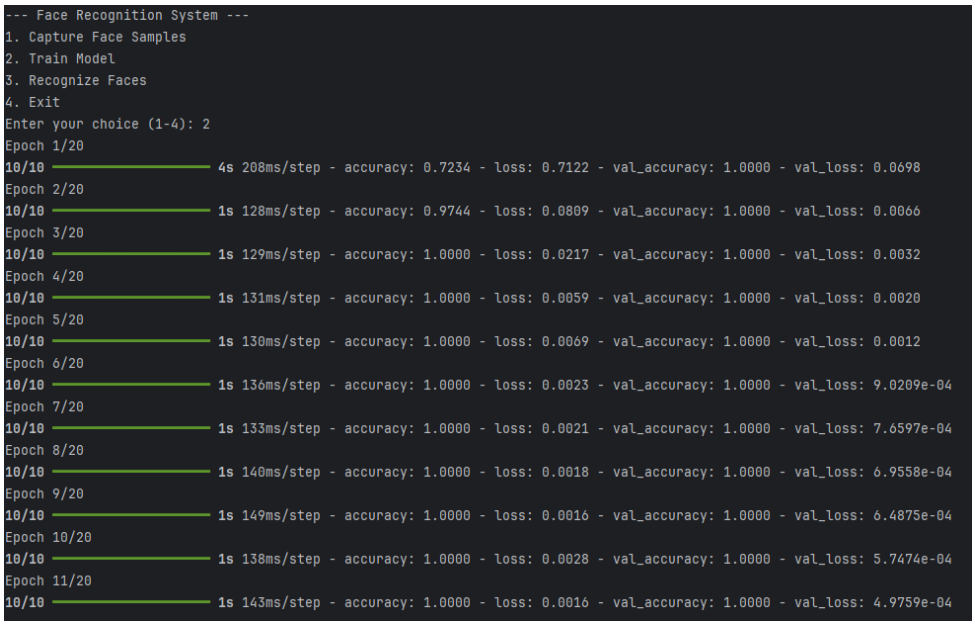
(a) Capture d'un nouveau visage

```
--- Face Recognition System ---
1. Capture Face Samples
2. Train Model
3. Recognize Faces
4. Exit
Enter your choice (1-4): 1
Enter person's name: Dhia
2024-12-19 19:08:04,020 - INFO: Captured 1/100
2024-12-19 19:08:04,063 - INFO: Captured 2/100
2024-12-19 19:08:04,079 - INFO: Captured 3/100
2024-12-19 19:08:04,113 - INFO: Captured 4/100
2024-12-19 19:08:04,183 - INFO: Captured 5/100
2024-12-19 19:08:04,185 - INFO: Captured 6/100
2024-12-19 19:08:04,212 - INFO: Captured 7/100
2024-12-19 19:08:04,214 - INFO: Captured 8/100
2024-12-19 19:08:04,240 - INFO: Captured 9/100
2024-12-19 19:08:04,273 - INFO: Captured 10/100
2024-12-19 19:08:04,305 - INFO: Captured 11/100
2024-12-19 19:08:04,337 - INFO: Captured 12/100
2024-12-19 19:08:04,368 - INFO: Captured 13/100
2024-12-19 19:08:04,392 - INFO: Captured 14/100
2024-12-19 19:08:04,428 - INFO: Captured 15/100
2024-12-19 19:08:04,483 - INFO: Captured 16/100
2024-12-19 19:08:04,523 - INFO: Captured 17/100
2024-12-19 19:08:04,544 - INFO: Captured 18/100
2024-12-19 19:08:04,545 - INFO: Captured 19/100
2024-12-19 19:08:04,583 - INFO: Captured 20/100
2024-12-19 19:08:04,585 - INFO: Captured 21/100
2024-12-19 19:08:04,613 - INFO: Captured 22/100
2024-12-19 19:08:04,643 - INFO: Captured 23/100
2024-12-19 19:08:04,681 - INFO: Captured 24/100
2024-12-19 19:08:04,703 - INFO: Captured 25/100
2024-12-19 19:08:04,736 - INFO: Captured 26/100
```

(b) Résultat de la capture dans le terminal

FIGURE 2 – Processus de capture faciale

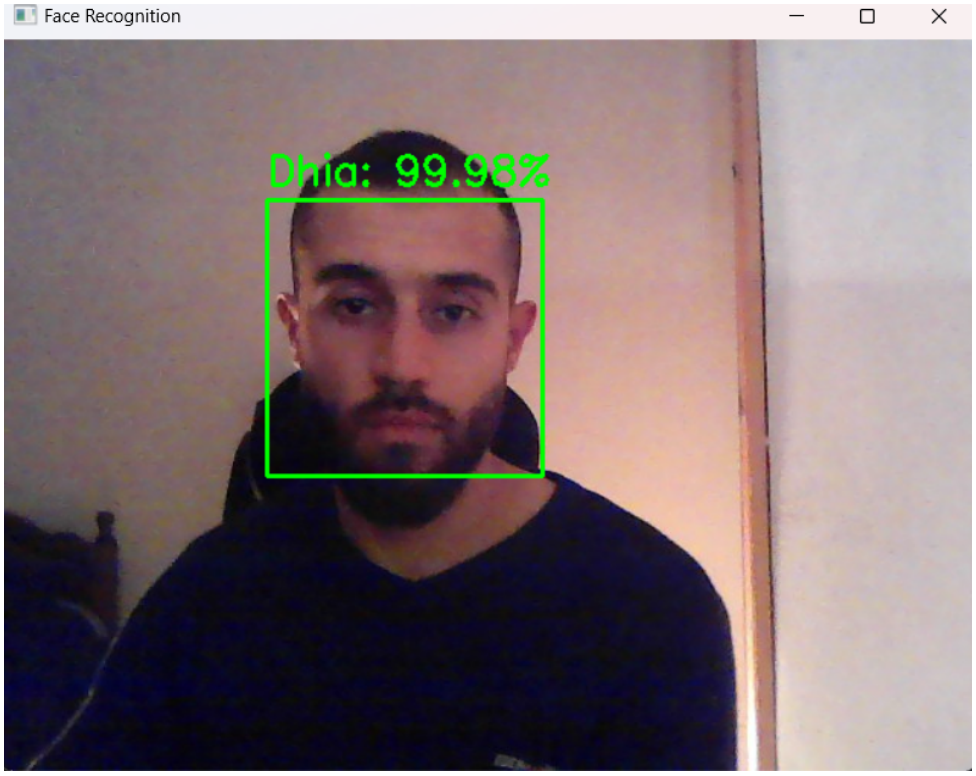
4.3 Entraînement du Modèle



(a) Entraînement du modèle avec les données capturées

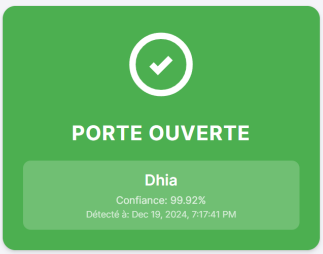
FIGURE 3 – Interface d'entraînement du modèle

4.4 Détection Faciale et Réponse Positive



(a) Utilisateur détecté

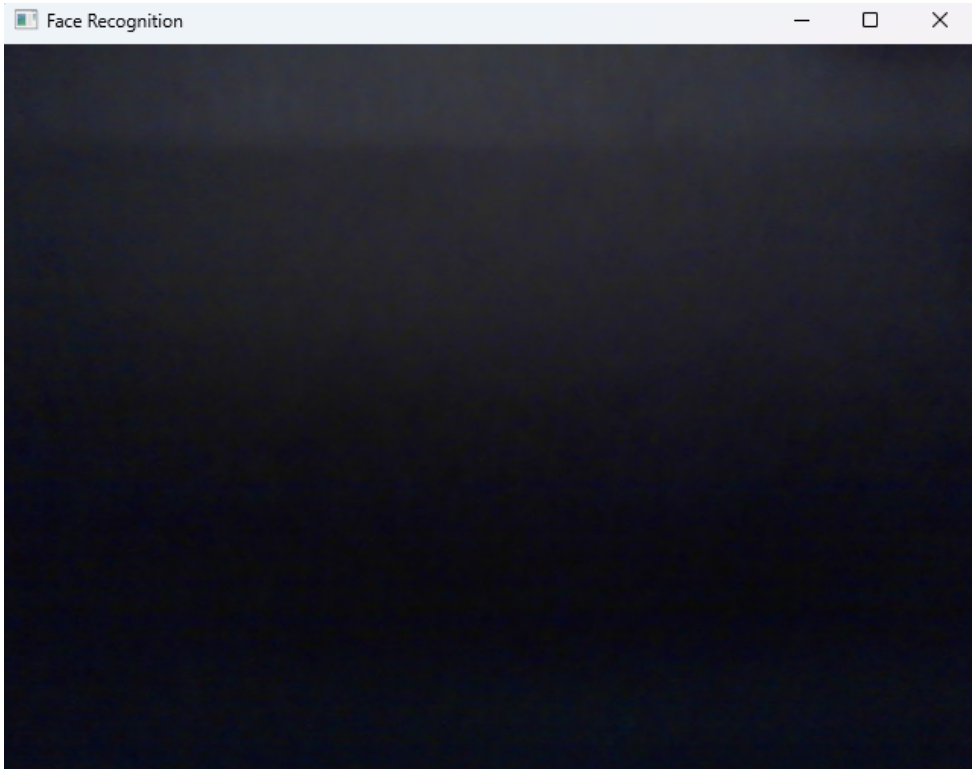
(b) Terminal lors de la détection faciale



(c) Réponse de l'interface lors de la détection

FIGURE 4 – Processus de détection positive

4.5 Détection Faciale et Réponse Négative



(a) Absence de personne devant la caméra

(b) Réponse de l'interface en absence de détection

FIGURE 5 – Processus de détection négative

4.6 Analyse des Performances

Les tests de démonstration ont été effectués dans diverses conditions :
— Éclairage variable (jour/nuit)
— Angles différents d'approche
— Multiples utilisateurs simultanés
— Tentatives d'usurpation d'identité

5 Résultats et Performances

5.1 Fonctionnalités Démontrées

— Reconnaissance faciale avec une précision élevée
— Contrôle d'accès automatique fiable
— Communication IoT en temps réel

5.2 Limites et Améliorations Potentielles

— Amélioration de la précision de reconnaissance dans des conditions extrêmes
— Optimisation de la gestion des conditions d'éclairage variables
— Intégration de mécanismes de sécurité avancés

6 Conclusion

Ce mini-projet démontre la faisabilité d'un système de reconnaissance faciale pour le contrôle d'accès intelligent. Il illustre l'intégration réussie des technologies de deep learning, de vision par ordinateur et d'IoT dans un contexte pratique de sécurité.

Les perspectives futures incluent :
— Amélioration des algorithmes de reconnaissance
— Intégration de fonctionnalités de sécurité supplémentaires
— Optimisation des performances système

7 Références

— Documentation OpenCV
— Documentation Face Recognition (Python)
— Documentation Paho MQTT
— Documentation Angular CLI