# Application de Détection Interactive

Documentation Technique et Guide d'Utilisation

Préparé par zeggane walid

12 avril 2025

## Table des matières

**1 Introduction** **2**
- 1.1 Présentation du logiciel . . . 2
- 1.2 Fonctionnalités principales . . . 2
- 1.3 Technologies utilisées . . . 2

**2 Architecture du logiciel** **2**
- 2.1 Structure globale . . . 2
- 2.2 Diagramme de classes . . . 2

**3 Détail des fonctionnalités** **3**
- 3.1 Détection des expressions faciales . . . 3
- 3.2 Détection des gestes de mains . . . 3

**4 Guide d'utilisation** **3**
- 4.1 Installation des prérequis . . . 3
- 4.2 Lancement de l'application . . . 3
- 4.3 Utilisation de l'interface . . . 4
  - 4.3.1 Écran d'accueil . . . 4
  - 4.3.2 Mode de détection . . . 4

**5 Aspects techniques** **4**
- 5.1 Traitement des images . . . 4
- 5.2 Algorithmes de détection . . . 4
  - 5.2.1 Détection d'expressions faciales . . . 4
  - 5.2.2 Détection de gestes de mains . . . 4

**6 Limitations actuelles et perspectives** **5**
- 6.1 Limitations . . . 5
- 6.2 Améliorations possibles . . . 5

**7 Transformation en application distribuable** **5**
- 7.1 Création d'un exécutable avec PyInstaller . . . 5
- 7.2 Création d'un installateur . . . 5
- 7.3 Distribution via des plateformes d'applications . . . 5

**8 Conclusion** **5**

**A Références** **6**

# 1 Introduction

## 1.1 Présentation du logiciel

L'Application de Détection Interactive est un logiciel qui combine la vision par ordinateur et l'intelligence artificielle pour détecter en temps réel les expressions faciales et les gestes des mains d'un utilisateur via la webcam de son ordinateur. Conçue comme une démonstration des possibilités offertes par les bibliothèques de traitement d'images modernes, cette application permet une interaction naturelle homme-machine par reconnaissance visuelle.

## 1.2 Fonctionnalités principales

— Détection d'expressions faciales en temps réel (sourire, surprise, colère)
— Reconnaissance de gestes de mains (OK, pouce levé, coeur, "I Love You")
— Interface graphique intuitive pour sélectionner différents modes de détection
— Visualisation superposée des points de repère du visage et des mains
— Affichage en temps réel des expressions et gestes détectés

## 1.3 Technologies utilisées

Le logiciel repose sur plusieurs technologies clés :
— **Python** comme langage de programmation principal
— **OpenCV** pour la capture vidéo et le traitement d'images
— **MediaPipe** de Google pour la détection précise des points de repère faciaux et manuels
— **Tkinter** pour l'interface graphique utilisateur
— **NumPy** pour les calculs mathématiques et le traitement des données

# 2 Architecture du logiciel

## 2.1 Structure globale

L'application est construite selon une architecture orientée objet avec une classe principale `DetectionApp` qui encapsule toutes les fonctionnalités. Elle est organisée en plusieurs composants fonctionnels :
— **Interface utilisateur** : création et gestion de l'écran d'accueil avec Tkinter
— **Capture vidéo** : initialisation et gestion du flux de la webcam avec OpenCV
— **Détection faciale** : analyse des points du visage avec MediaPipe Face Mesh
— **Détection des mains** : analyse des gestes manuels avec MediaPipe Hands
— **Visualisation** : affichage des résultats sur le flux vidéo en temps réel

## 2.2 Diagramme de classes

La classe principale `DetectionApp` contient les méthodes suivantes :
— `__init__` : Initialise l'application et ses composants
— `create_welcome_screen` : Crée l'interface graphique d'accueil
— `exit_application` : Gère la fermeture propre de l'application
— `detect_facial_expression` : Analyse les expressions du visage
— `detect_hand_gesture` : Analyse les gestes des mains

— `start_detection` : Lance le mode de détection sélectionné par l'utilisateur

# 3 Détail des fonctionnalités

## 3.1 Détection des expressions faciales

La détection des expressions faciales s'appuie sur l'analyse géométrique des 468 points de repère fournis par MediaPipe Face Mesh. L'application utilise ces points pour calculer diverses mesures relatives à :

— **La bouche** : ratio hauteur/largeur, position des coins
— **Les sourcils** : distance par rapport à une position neutre
— **Les yeux** : degré d'ouverture, position par rapport au visage

Les expressions reconnues sont :

— **Sourire** : détecté lorsque les coins de la bouche sont relevés et que la bouche est large
— **Surprise** : détectée par des sourcils relevés et une bouche ouverte (grand ratio hauteur/largeur)
— **Colère** : identifiée par des sourcils abaissés et froncés

L'algorithme compare les valeurs calculées à des seuils prédéfinis pour décider quelle expression est présente.

## 3.2 Détection des gestes de mains

La détection des gestes de mains utilise les 21 points de repère fournis par MediaPipe Hands, correspondant aux articulations et extrémités des doigts. L'application calcule les angles, distances et positions relatives entre ces points pour identifier des gestes spécifiques :

— **OK** : pouce et index formant un cercle, autres doigts tendus
— **Like (pouce levé)** : pouce vers le haut, autres doigts repliés
— **Cœur** : pouce et index tendus et rapprochés, autres doigts repliés
— **I Love You** : pouce, index et auriculaire tendus, autres doigts repliés

L'algorithme prend aussi en compte la différence entre main gauche et main droite pour une détection précise.

# 4 Guide d'utilisation

## 4.1 Installation des prérequis

Pour utiliser l'application, les bibliothèques suivantes doivent être installées :

```
pip install opencv-python
pip install mediapipe
pip install numpy
pip install pillow
```

Tkinter est généralement inclus dans l'installation standard de Python.

## 4.2 Lancement de l'application

Pour lancer l'application, exécutez simplement le script principal :

```
python detection_app.py
```

## 4.3 Utilisation de l'interface

### 4.3.1 Écran d'accueil

À l'ouverture de l'application, vous êtes accueilli par un écran présentant trois options :
— **Détection du Visage** : active uniquement la reconnaissance des expressions faciales
— **Détection des Mains** : active uniquement la reconnaissance des gestes de mains
— **Les Deux** : active simultanément les deux modes de détection

### 4.3.2 Mode de détection

Une fois un mode sélectionné :
— Le flux vidéo de votre webcam s'affiche en plein écran
— Les points de repère (visage ou mains) sont superposés en semi-transparence
— Les expressions ou gestes détectés s'affichent en grand texte à l'écran
— Appuyez sur 'Q' à tout moment pour quitter et revenir à l'écran d'accueil

# 5 Aspects techniques

## 5.1 Traitement des images

L'application effectue plusieurs opérations sur chaque image capturée :
— Inversion horizontale pour créer un effet miroir
— Conversion de BGR (format OpenCV) à RGB (format MediaPipe)
— Création d'une image de fond noir pour dessiner les points de repère
— Combinaison des images originale et annotée avec transparence
— Ajout de texte pour afficher les instructions et les détections

## 5.2 Algorithmes de détection

### 5.2.1 Détection d'expressions faciales

L'algorithme utilise les distances relatives et ratios entre points clés :

```
# Exemple simplifi  pour la d tection du sourire
mouth_aspect_ratio = mouth_height / mouth_width
if mouth_aspect_ratio > smile_threshold and
   mouth_corner_y_avg < bottom_lip[1]:
    return "SOURIRE :)"
```

### 5.2.2 Détection de gestes de mains

L'algorithme vérifie la position de chaque doigt et leurs relations :

```
# Exemple simplifi  pour le geste "Like"
is_thumb_up = thumb_tip.y < thumb_ip.y
fingers_closed = (index_tip.y > index_pip.y and ...)
if is_thumb_up and fingers_closed:
    return "LIKE       "
```

# 6 Limitations actuelles et perspectives

## 6.1 Limitations

— La détection peut être sensible aux conditions d'éclairage
— Les seuils de détection sont fixes et peuvent ne pas convenir à tous les utilisateurs
— La reconnaissance est limitée à 3 expressions faciales et 4 gestes de mains
— Une seule personne peut être détectée à la fois

## 6.2 Améliorations possibles

— Ajout de plus d'expressions et gestes à reconnaître
— Implémentation d'un système d'apprentissage pour améliorer la précision
— Ajout d'un système de calibration personnalisée par utilisateur
— Support de multiples personnes dans le champ de vision
— Création d'applications interactives basées sur les gestes (jeux, contrôle d'interface...)
— Support de nouvelles plateformes (mobile, web) via des wrappers ou une réimplémentation

# 7 Transformation en application distribuable

Pour convertir ce logiciel en une application autonome que les utilisateurs peuvent installer et exécuter directement, plusieurs approches sont possibles :

## 7.1 Création d'un exécutable avec PyInstaller

PyInstaller permet de regrouper le script Python et toutes ses dépendances dans un seul fichier exécutable :

```
pip install pyinstaller
pyinstaller --onefile --windowed detection_app.py
```

## 7.2 Création d'un installateur

Pour créer un véritable installateur, des outils supplémentaires peuvent être utilisés :
— NSIS (Nullsoft Scriptable Install System) pour Windows
— pkgbuild/productbuild pour macOS
— Debian Package Builder pour Linux

## 7.3 Distribution via des plateformes d'applications

Après avoir créé un installateur, l'application peut être distribuée via :
— Microsoft Store pour Windows
— Mac App Store pour macOS
— Dépôts Linux (PPA, etc.)

# 8 Conclusion

L'Application de Détection Interactive démontre les possibilités offertes par les technologies modernes de vision par ordinateur rendues accessibles par des bibliothèques comme

MediaPipe et OpenCV. En combinant ces outils avec une interface utilisateur intuitive, l'application offre une expérience engageante qui peut servir de base à de nombreuses applications pratiques dans des domaines comme l'interaction homme-machine, les jeux, l'accessibilité ou l'éducation.

Le code modularisé et bien commenté facilite l'extension de ses fonctionnalités pour des cas d'usage spécifiques ou l'intégration dans des projets plus importants.

# A Références


— Documentation OpenCV : https://docs.opencv.org/
— Documentation MediaPipe : https://google.github.io/mediapipe/
— Documentation Tkinter : https://docs.python.org/3/library/tk.html
— Guide PyInstaller : https://pyinstaller.org/en/stable/