

# **Οδηγός Εγκατάστασης και Ελέγχου Αναγνώρισης Προσώπων**

## Περιεχόμενα

1. Εισαγωγή .....	4
2. Εγκατάσταση των Εξαρτήσεων .....	4
2.1 Python .....	4
2.2 pip .....	5
2.3 CMake.....	5
2.4 Microsoft Visual Studio Build Tools .....	6
2.5 Εγκατάσταση OpenCV .....	7
2.6 Εγκατάσταση SciPy.....	7
2.7 dlib .....	8
2.8 Κατεβάστε τα προ-εκπαιδευμένα μοντέλα .....	8
2.9 Δημιουργία φακέλου known_faces όπου περιέχει φακέλους με εικόνες προσώπων για αναγνώριση με την εξής δομή: .....	8
3. Έλεγχος Εγκατάστασης .....	9
3.1. Έλεγχος Python .....	9
3.2. Έλεγχος pip .....	9
3.3. Έλεγχος CMake.....	9
3.4. Έλεγχος Microsoft Visual Studio .....	10
3.5. Έλεγχος OpenCV .....	10
3.6. Έλεγχος SciPy .....	10
3.7. Έλεγχος dlib .....	10
3.8. Έλεγχος για τη βιβλιοθήκη Face Recognition.....	11
4. Εφαρμογή Αναγνώρισης Προσώπων.....	11
4.1. Κώδικας .....	11
4.2. Ανάλυση του κώδικα .....	13
4.3. Οδηγίες εκτέλεσης στο PowerShell.....	14
5. Περιορισμοί του προγράμματος .....	15
5.1. Θόρυβος στην εικόνα ή στο βίντεο .....	15
5.2. Χαμηλή ανάλυση εισόδου .....	15
5.3. Ακρίβεια του μοντέλου αναγνώρισης.....	15
5.4. Θέση και γωνία προσώπου .....	15
5.5. Πολυπλοκότητα σκηνών .....	16
5.6. Επεξεργαστική Ικανότητα.....	16
5.7. Περιορισμοί στην επεξεργασία.....	16
6. Προτάσεις για βελτίωση .....	16
1. Χρήση εικόνων υψηλής ποιότητας.....	16

2.	Καλή φωτεινότητα και εστίαση .....	16
3.	Αύξηση ανοχής (tolerance).....	16
4.	Βελτιστοποίηση απόδοσης.....	17
	Γιατί να περιορίσεις την ανάλυση εισόδου ή τα FPS; .....	17
	<ul style="list-style-type: none"> <li>Υψηλή Ανάλυση: Όσο υψηλότερη είναι η ανάλυση του βίντεο, τόσο περισσότερα δεδομένα πρέπει να επεξεργαστεί το σύστημα. Αυτό αυξάνει το χρόνο επεξεργασίας ανά καρέ.....</li> <li>Υψηλά FPS: Όταν το βίντεο έχει υψηλά καρέ ανά δευτερόλεπτο, το πρόγραμμα πρέπει να επεξεργάζεται περισσότερα καρέ ανά δευτερόλεπτο, γεγονός που επιβαρύνει την CPU ή την GPU.....</li> </ul>	17
	Πώς να περιορίσεις την ανάλυση εισόδου; .....	17
	Πώς να περιορίσεις την ανάλυση εισόδου; .....	17
	Συνδυασμός μείωσης ανάλυσης και FPS.....	18
	Οφέλη της Βελτιστοποίησης.....	19
	Νεός κωδικός προσαρμοσμένος .....	19
7.	Συμπεράσματα .....	24
7..1.	Βασικά σημεία:.....	24
7..2.	Τελική Εκτίμηση .....	25
7..3.	Πιο Σύνθετες Εφαρμογές Αναγνώρισης Προσώπων.....	25

# 1.Εισαγωγή

Σε αυτόν τον οδηγό, περιγράφονται βήμα-βήμα οι διαδικασίες για την εγκατάσταση, τον έλεγχο και τη χρήση ενός συστήματος αναγνώρισης προσώπων με Python. Το σύστημα χρησιμοποιεί τις βιβλιοθήκες dlib και face\_recognition για την επεξεργασία εικόνων και βίντεο.

Το project περιλαμβάνει:

- Εγκατάσταση βασικών βιβλιοθηκών και εργαλείων.
- Έλεγχο της ορθότητας των εγκαταστάσεων.
- Υλοποίηση της ανίχνευσης και αναγνώρισης προσώπων.

## 2.Εγκατάσταση των Εξαρτήσεων

### 2.1 Python

Η Python είναι απαραίτητη για την υλοποίηση του συστήματος.

**Ενέργειες:**

1. Κατεβάστε την Python 3.11.9 από την επίσημη ιστοσελίδα:  
<https://www.python.org/downloads/>.
2. Κατά την εγκατάσταση, ενεργοποιήστε την επιλογή "Add Python to PATH".
3. Ελέγξτε την εγκατάσταση με την εντολή:  
`python --version`

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται η έκδοση της Python, π.χ., Python 3.11.9

4. Σε περίπτωση που δεν εμφανίζεται το παραπάνω μήνυμα πληκτρολογήστε στο Power shell:

`C:\Users\xxx\AppData\Local\Programs\Python\Python311\python.exe --version`

**Αναμενόμενο αποτέλεσμα:** Να εμφανιστεί η έκδοση της Python, Python 3.11.9 , τότε το πρόβλημα αφορά τη ρύθμιση του PATH.

5. Διαχείριση Μεταβλητών Περιβάλλοντος
  - 5..1. Πατήστε Win + S και αναζητήστε Environment Variables.
  - 5..2. Κάντε κλικ στο Edit the system environment variables.
  - 5..3. Στο παράθυρο System Properties, πατήστε Environment Variables.
  - 5..4. Στη λίστα System variables, βρείτε τη μεταβλητή Path και επιλέξτε Edit.
  - 5..5. Βεβαιωθείτε ότι οι παρακάτω διαδρομές υπάρχουν στη λίστα:
    - 5..5.1. C:\Users\xxx\AppData\Local\Programs\Python\Python311
    - 5..5.2. C:\Users\xxx\AppData\Local\Programs\Python\Python311\Scripts

## 6. Έλεγχος στο PATH

Εκτελέστε την εντολή στο Power Shell :

```
$env:Path -split ";"
```

**Αναμενόμενο αποτέλεσμα:** Βεβαιωθείτε ότι οι παραπάνω διαδρομές εμφανίζονται στο αποτέλεσμα.

## 2.2 pip

Το pip είναι ο διαχειριστής πακέτων της Python που επιτρέπει την εγκατάσταση βιβλιοθηκών. Το pip επι της ουσίας είναι το εργαλείο που σου επιτρέπει να εγκαταστήσεις βιβλιοθήκες που επεκτείνουν τη λειτουργικότητα της Python. Χωρίς το pip, η διαχείριση εξωτερικών βιβλιοθηκών θα ήταν δύσκολη.

### Ενέργειες:

1. Επισκεφτείτε το σύνδεσμο <https://bootstrap.pypa.io/get-pip.py>
2. Θα σας ανοίξει ένα κείμενο το οποίο πρέπει να το αντιγράψετε
3. Ανοίξτε έναν επεξεργαστή κειμένου (**Notepad**)
4. Επικολλήστε το περιεχόμενο
5. Αποθηκεύστε το αρχείο ως εξής
  - Επιλέξτε **File > Save As**.
  - Δώστε όνομα: get-pip.py.
  - Επιλέξτε τη θέση αποθήκευσης, C:\Project.
  - Αλλάξτε το **Save as type** σε **All Files** και αποθηκεύστε με κατάληξη .py
6. Ελέγξτε αν το pip είναι εγκατεστημένο:  

```
pip --version
```

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται η έκδοση του pip, pip 24.3.1 from /usr/local/lib/python3.11/site-packages.

7. Ελέγξτε που είναι εγκατεστημένο το pip με την εντολή στο Power Shell:
  - C:\WINDOWS\system32> py -m ensurepip --upgrade
  - C:\WINDOWS\system32> py -m pip --version (ενναλλακτικά)

**Αναμενόμενο αποτέλεσμα:** Τη θέση και την έκδοση του PIP

8. Εγκαταστήστε τις απαραίτητες βιβλιοθήκες με την παρακάτω εντολή στο Power Shell:

```
PS C:\Project> py -m pip install face_recognition
```

**Αναμενόμενο αποτέλεσμα:** Μηνύματα επιτυχούς εγκατάστασης των βιβλιοθηκών.

## 2.3 CMake

Το CMake χρησιμοποιείται για τη διαμόρφωση και μεταγλώττιση λογισμικού.

### Ενέργειες:

1. Εγκαταστήστε το CMake:  
`pip install cmake`

**Αναμενόμενο αποτέλεσμα:** Μήνυμα επιτυχούς εγκατάστασης, π.χ., Successfully installed cmake-3.31.2.

2. Ελέγξτε την εγκατάσταση:  
`cmake --version`

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται η έκδοση του CMake, π.χ., cmake version 3.31.2.

## 2.4 Microsoft Visual Studio Build Tools

Το Microsoft Visual Studio είναι απαραίτητο για τη μεταγλώττιση του dlib.

### Ενέργειες:

1. Κατεβάστε το από την επίσημη ιστοσελίδα της Microsoft:  
<https://visualstudio.microsoft.com/downloads/?q=build+tools>.
2. Κατά την εγκατάσταση (Modify) ελέγξτε να:
  - 2.1. Έχετε επιλέξει Desktop development with C++
  - 2.1. Βεβαιώσου ότι έχεις επιλέξει τα MSVC Compiler και Windows 10 SDK
  - 2.1. Επίλεξε το εργαλείο CMake
  - 2.1. Επανεκκίνησε τον υπολογιστή σου
3. Ελέγξτε την εγκατάσταση με την εντολή στο Power shell:  
`cl`

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται το μήνυμα Microsoft (R) C/C++ Optimizing Compiler.

4. Σε περίπτωση που δεν εμφανίσει το παραπάνω μήνυμα και εμφανίσει το παρακάτω:

```
C:\WINDOWS\system32> cl
```

```
cl : The term 'cl' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
```

```
At line:1 char:1
```

```
+ cl
```

```
+ ~~
```

```
+ CategoryInfo: ObjectNotFound: (cl:String) [],
```

```
CommandNotFoundException
```

```
+ FullyQualifiedErrorId : CommandNotFoundException
```

Το μήνυμα αυτό δείχνει ότι η εντολή `cl` δεν αναγνωρίζεται και υπάρχει πιθανότητα η διαδρομή (PATH) για το Visual Studio Compiler να μην έχει ρυθμιστεί σωστά στο περιβάλλον σας. Εκτελέστε τα παρακάτω βήματα ώστε να φτάσετε στο παραπάνω αναμενόμενο αποτέλεσμα.

### Άνοιγμα του Developer Command Prompt:

- a) Πατήστε το κουμπί **Start** (Εναρξη).
- b) Πληκτρολογήστε και ανοίξτε ως διαχειριστής : **Developer Command Prompt for Visual Studio 2022**
- c) Αφού ανοίξετε το Developer Command Prompt, πληκτρολογήστε: **cl**
- d) **Αναμενόμενο Αποτέλεσμα:** Εμφανίζεται ένα μήνυμα παρόμοιο με το παρακάτω.

**Microsoft (R) C/C++ Optimizing Compiler Version 19.42.34436  
for x86**

**Copyright (C) Microsoft Corporation. All rights reserved.**

**usage: cl [ option... ] filename... [ /link linkoption... ]**

## 2.5 Εγκατάσταση OpenCV

Το OpenCV χρησιμοποιείται για την επεξεργασία εικόνων και βίντεο.

### Ενέργειες:

1. Εγκαταστήστε το OpenCV πληκτρολογώντας στο Power Shell:  
`pip install opencv-python`

**Αναμενόμενο αποτέλεσμα:** Μήνυμα επιτυχούς εγκατάστασης.

2. Επιβεβαιώστε την εγκατάσταση αφού ανοίξουμε περιβάλλον Python:  
`>>>import cv2`  
`>>>print(cv2.__version__)`

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται η έκδοση του OpenCV, π.χ., 4.5.3.

## 2.6 Εγκατάσταση SciPy

Το SciPy χρησιμοποιείται για υποστήριξη υπολογισμών και στατιστικής ανάλυσης.

### Ενέργειες:

1. Εγκαταστήστε το SciPy:  
`pip install scipy`

**Αναμενόμενο αποτέλεσμα:** Μήνυμα επιτυχούς εγκατάστασης.

3. Επιβεβαιώστε την εγκατάσταση αφού ανοίξουμε περιβάλλον Python:  
`>>>import scipy`  
`>>>print(scipy.__version__)`

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται η έκδοση του SciPy, π.χ., 1.9.1

## 2.7 dlib

Η βιβλιοθήκη dlib χρησιμοποιείται για αναγνώριση και ανίχνευση προσώπων.

### Ενέργειες:

1. Εγκαταστήστε το dlib:  
`pip install dlib --no-binary dlib`

**Αναμενόμενο αποτέλεσμα:** Μηνύματα που δείχνουν την επιτυχή μεταγλώττιση και εγκατάσταση.

## 2.8 Κατεβάστε τα προ-εκπαιδευμένα μοντέλα

1. `shape_predictor_68_face_landmarks.dat`
  - Χρησιμοποιείται για την ανίχνευση χαρακτηριστικών προσώπου.
  - Από τον ιστότοπο:  
[http://dlib.net/files/shape\\_predictor\\_68\\_face\\_landmarks.dat.bz2](http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2)
2. `dlib_face_recognition_resnet_model_v1.dat`
  - Χρησιμοποιείται για την αναγνώριση προσώπων.
  - Από τον ιστότοπο:  
[http://dlib.net/files/dlib\\_face\\_recognition\\_resnet\\_model\\_v1.dat.bz2](http://dlib.net/files/dlib_face_recognition_resnet_model_v1.dat.bz2)
3. Τοποθετήστε τα αρχεία στον φάκελο `models` του `project` με την μορφή `C:\Project\models` και αποσυμπίστετα.

## 2.9 Δημιουργία φακέλου `known_faces` όπου περιέχει φακέλους με εικόνες προσώπων για αναγνώριση με την εξής δομή:

1. Φάκελος `known_faces`

```
C:/Project/known_faces/  
├── Person1/  
│   ├── image1.jpg  
│   └── image2.jpg  
└── Person2/  
    ├── image1.jpg  
    └── image2.jpg
```

2. Βίντεο

Το πρόγραμμα δέχεται βίντεο από τον φάκελο:  
`C:/Project/video/export_1713300134091.MP4`



3. Επιβεβαιώστε τη δομή φακέλου:

Ο φάκελος C:/Project/ στον νέο υπολογιστή πρέπει να έχει την παρακάτω δομή:

```
C:/Project/
├── models/
│   ├── shape_predictor_68_face_landmarks.dat
│   └── dlib_face_recognition_resnet_model_v1.dat
├── known_faces/
│   ├── Person1/
│   │   ├── image1.jpg
│   │   └── image2.jpg
│   ├── Person2/
│   │   ├── image1.jpg
│   │   └── image2.jpg
├── video/
│   └── export_1713300134091.MP4
└── face_detection_bw.py
```

## 3. Έλεγχος Εγκατάστασης

Αφού ολοκληρώσετε τις εγκαταστάσεις, πρέπει να **επαληθεύσετε** ότι όλα λειτουργούν σωστά.

### 3.1. Έλεγχος Python

Ελέγξτε αν το pip είναι εγκατεστημένο με την εντολή:  
python -version

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται η έκδοση της Python, π.χ., Python 3.11.9.

### 3.2. Έλεγχος pip

Ελέγξτε αν το pip είναι εγκατεστημένο με την εντολή:  
pip -version

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται η έκδοση του pip

### 3.3. Έλεγχος CMake

Ελέγξτε εάν το CMake είναι εγκατεστημένο με την εντολή:  
cmake -version

**Αναμενόμενο αποτέλεσμα:** Εμφανίζεται η έκδοση του CMake

### 3.4. Έλεγχος Microsoft Visual Studio

Ελέγξτε εάν το Microsoft Visual Studio είναι εγκατεστημένο με την εντολή:  
CI

Αναμενόμενο αποτέλεσμα: Εμφανίζεται το μήνυμα Microsoft (R) C/C++ Optimizing Compiler.

### 3.5. Έλεγχος OpenCV

Ελέγξτε εάν το Open CV είναι εγκατεστημένο με την εντολή στο Power Shell:  
Python και μετά το μήνυμα

Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>>

Τις παρακάτω εντολές:

```
import cv2  
print(cv2.__version__)
```

Αναμενόμενο αποτέλεσμα: Εμφανίζεται η έκδοση του OpenCV, π.χ., 4.10.0.

### 3.6. Έλεγχος SciPy

Ελέγξτε εάν το SciPy είναι εγκατεστημένο με την εντολή στο Power Shell:  
Python και μετά το μήνυμα

Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>>

Τις παρακάτω εντολές:

```
import scipy  
print(scipy.__version__)
```

Αναμενόμενο αποτέλεσμα: Εμφανίζεται η έκδοση του SciPy, π.χ., 1.14.1

### 3.7. Έλεγχος dlib

Ελέγξτε αν το dlib είναι εγκατεστημένο με την εντολή στο Power Shell:

Python και μετά το μήνυμα

Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>>

```
Τις παρακάτω εντολές:  
>>>import dlib print  
>>>(dlib.__version__)  
>>>print(dlib.DLIB_USE_CUDA)
```

**Αναμενόμενο αποτέλεσμα:** Εμφάνιση της έκδοσης του DLIP 19.24.6 και True αν υποστηρίζεται η GPU για το DLIP, διαφορετικά False.

### 3.8. Έλεγχος για τη βιβλιοθήκη Face Recognition

Ελέγξτε αν η βιβλιοθήκη Face Recognition είναι εγκατεστημένο με την εντολή στο Power Shell:

Python και μετά το μήνυμα

```
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64  
bit (AMD64)] on win32
```

Type "help", "copyright", "credits" or "license" for more information.

```
>>>
```

Τις παρακάτω εντολές:

```
import face_recognition  
print(face_recognition.__version__)
```

**Αναμενόμενο αποτέλεσμα:** Εμφάνιση την έκδοση της βιβλιοθήκης, 1.2.3

## 4.Εφαρμογή Αναγνώρισης Προσώπων

Ο παρακάτω κώδικας υλοποιεί ένα σύστημα ανίχνευσης προσώπων:

### 4.1. Κώδικας

```
import cv2  
import face_recognition  
import os  
  
# Έλεγχος για τα αρχεία .dat  
MODEL_PATH = "models"  
SHAPE_PREDICTOR_PATH=os.path.join(MODEL_PATH,"shape_predictor_68_f  
ace_landmarks.dat")  
FACE_RECOGNITION_MODEL_PATH=os.path.join(MODEL_PATH,  
"dlib_face_recognition_resnet_model_v1.dat")  
  
If not os.path.exists(SHAPE_PREDICTOR_PATH) or not  
os.path.exists(FACE_RECOGNITION_MODEL_PATH):  
    print("Τα απαραίτητα αρχεία δεν βρέθηκαν.")  
    exit()  
  
print("Τα αρχεία βρέθηκαν επιτυχώς!")  
  
# Φόρτωση γνωστών προσώπων
```

```

KNOWN_FACES_PATH = "known_faces"
if not os.path.exists(KNOWN_FACES_PATH):
    print("Ο φάκελος γνωστών προσώπων δεν βρέθηκε.")
    exit()

known_encodings = []
known_names = []

# Για κάθε φάκελο (όνομα ατόμου) στον φάκελο known_faces
for person_name in os.listdir(KNOWN_FACES_PATH):
    person_folder = os.path.join(KNOWN_FACES_PATH, person_name)

    if not os.path.isdir(person_folder):
        continue # Αγνόησε αρχεία που δεν είναι φάκελοι

    # Επεξεργασία κάθε εικόνας στον φάκελο
    for image_name in os.listdir(person_folder):
        image_path = os.path.join(person_folder, image_name)
        image = face_recognition.load_image_file(image_path)
        encodings = face_recognition.face_encodings(image)

        if len(encodings) > 0:
            known_encodings.append(encodings[0])
            known_names.append(person_name) # Το όνομα είναι το όνομα του
            φακέλου

print(f'Φορτώθηκαν {len(known_encodings)} γνωστά πρόσωπα από πολλαπλές
εικόνες.")

# Επιλογή εισόδου
choice = input("Πατήστε 1 για κάμερα ή 2 για αποθηκευμένο βίντεο: ")
if choice == "1":
    video_source = 0 # Κάμερα
elif choice == "2":
    video_source = "C:/Project/video/export_1713300134091.MP4".strip()
    if not os.path.exists(video_source):
        print(f"Το αρχείο βίντεο δεν βρέθηκε στο μονοπάτι: {video_source}")
        exit()
else:
    print("Μη έγκυρη επιλογή.")
    exit()

# Έναρξη βίντεο
video_capture = cv2.VideoCapture(video_source)

while video_capture.isOpened():
    ret, frame = video_capture.read()
    if not ret:
        print("Δεν είναι δυνατή η ανάγνωση από την πηγή.")
        break

    # Μετατροπή του καρέ σε ασπρόμαυρο
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Ανίχνευση προσώπων

```

```

face_locations = face_recognition.face_locations(gray_frame)
face_encodings = face_recognition.face_encodings(frame, face_locations)

# Μετατροπή του ασπρόμαυρου καρέ σε έγχρωμο για σχεδιασμό κόκκινων
πλαισίων
color_frame = cv2.cvtColor(gray_frame, cv2.COLOR_GRAY2BGR)

# Σχεδίαση πλαισίων και ονομάτων γύρω από τους στόχους
for face_location, face_encoding in zip(face_locations, face_encodings):
    matches = face_recognition.compare_faces(known_encodings, face_encoding)
    name = "Άγνωστος"

    if True in matches:
        match_index = matches.index(True)
        name = known_names[match_index] # Όνομα από το φάκελο του ατόμου

    top, right, bottom, left = face_location
    # Κόκκινο περίγραμμα
    cv2.rectangle(color_frame, (left, top), (right, bottom), (0, 0, 255), 2)
    # Διαφανές κόκκινο πλαίσιο
    overlay = color_frame.copy()
    cv2.rectangle(overlay, (left, top), (right, bottom), (0, 0, 255), -1)
    alpha = 0.3
    color_frame = cv2.addWeighted(overlay, alpha, color_frame, 1 - alpha, 0)

# Εμφάνιση ονόματος πάνω από το πλαίσιο
cv2.putText(
    color_frame,
    name,
    (left, top - 10), # Το όνομα θα εμφανίζεται πάνω από το πλαίσιο
    cv2.FONT_HERSHEY_SIMPLEX,
    0.8,
    (0, 0, 255),
    2
)

# Προβολή βίντεο
cv2.imshow("Αναγνώριση Προσώπων", color_frame)

# Έξοδος με 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Καθαρισμός
video_capture.release()
cv2.destroyAllWindows()

```

## 4.2. Ανάλυση του κώδικα

### 1. Αρχικοποίηση και έλεγχος αρχείων μοντέλων

- ο Ελέγχεται αν υπάρχουν δύο αρχεία:
  - `shape_predictor_68_face_landmarks.dat` (για ανίχνευση χαρακτηριστικών προσώπων)

- `dlib_face_recognition_resnet_model_v1.dat` (για αναγνώριση προσώπων).
  - ο Αν δεν βρεθούν, ο κώδικας σταματά και εμφανίζεται σχετικό μήνυμα.
- 2. **Φόρτωση γνωστών προσώπων**
  - ο Τα δεδομένα των γνωστών προσώπων αποθηκεύονται στον φάκελο `known_faces`.
  - ο Για κάθε φάκελο μέσα στο `known_faces`, αντιστοιχεί ένα όνομα προσώπου.
  - ο Οι εικόνες από αυτούς τους φακέλους φορτώνονται και εξάγονται τα χαρακτηριστικά τους μέσω της `face_recognition`.
- 3. **Επιλογή εισόδου**
  - ο Ο χρήστης επιλέγει αν θα χρησιμοποιηθεί κάμερα (1) ή αποθηκευμένο βίντεο (2).
  - ο Αν η επιλογή είναι 2, ελέγχεται αν το βίντεο υπάρχει.
- 4. **Επεξεργασία βίντεο**
  - ο Το πρόγραμμα διαβάζει κάθε καρέ του βίντεο.
  - ο Μετατρέπει κάθε καρέ σε ασπρόμαυρο για βελτίωση της ανίχνευσης.
  - ο Ανιχνεύονται πρόσωπα και συγκρίνονται με τα γνωστά δεδομένα.
  - ο Ταυτοποιούνται τα πρόσωπα και σχεδιάζονται πλαίσια με τα ονόματά τους.
- 5. **Έξοδος**
  - ο Πατώντας το πλήκτρο `q`, ο χρήστης κλείνει την εφαρμογή.

## 4.3. Οδηγίες εκτέλεσης στο PowerShell

### 1. Αποθηκεύστε τον κώδικα

- Άνοιξε έναν επεξεργαστή κειμένου (π.χ., Notepad) ή κατεβάστε το Microsoft Visual Studio code το οποίο μας βοηθάει να κάνουμε συγκρίσεις μεταξύ δυο κωδικών.
- Αντιγράψτε και επικολλήστε τον παραπάνω κώδικα ως New File.
- Αποθήκευσε το αρχείο ως όνομα αρχείου `"face_recognition_bw16.py"` (για τον παραπάνω κώδικα) και επιλέγοντας Αποθήκευση ως "All files" στον φάκελο `C:\Project`.

- **Εκτέλεση του κώδικα** Άνοιξε το PowerShell, πήγαινε στον φάκελο όπου βρίσκεται ο κώδικας και πληκτρολόγησε:

1.3.1.1.1. `cd C:\Project`

1.3.1.1.2. `PS C:\Project> py -3.11 face_detection_bw16.py`

1.3.1.1.3. Περιμένετε μέχρι να:

1.3.1.1.3.1. Επίλεξε 1 για την κάμερα ή 2 για βίντεο.

1.3.1.1.3.2. Πατώντας το πλήκτρο `q`, κλείνει η εφαρμογή.

## 5.Περιορισμοί του προγράμματος

### 5.1. Θόρυβος στην εικόνα ή στο βίντεο

- **Θόρυβος από χαμηλό φωτισμό:**
  - Σε συνθήκες χαμηλού φωτισμού, η κάμερα μπορεί να παράγει θορυβώδη καρέ, τα οποία δυσκολεύουν την ανίχνευση προσώπων.
  - Η μετατροπή σε ασπρόμαυρη εικόνα μπορεί να βελτιώσει την κατάσταση, αλλά αν ο θόρυβος είναι έντονος, η ακρίβεια πέφτει.
- **Κίνηση και θολά καρέ:**
  - Αν τα πρόσωπα κινούνται γρήγορα ή η κάμερα έχει κακή ποιότητα λήψης, τα καρέ μπορεί να είναι θολά, επηρεάζοντας την ακρίβεια της ανίχνευσης.

### 5.2. Χαμηλή ανάλυση εισόδου

- Η ανίχνευση και η αναγνώριση προσώπων βασίζονται στην ευκρίνεια της εικόνας.
  - Αν η ανάλυση είναι πολύ χαμηλή, τα χαρακτηριστικά του προσώπου (π.χ., μάτια, μύτη, στόμα) δεν μπορούν να αναγνωριστούν σωστά.
  - Για καλύτερα αποτελέσματα, χρησιμοποίησε εικόνες ή βίντεο με ανάλυση τουλάχιστον 720p.

### 5.3. Ακρίβεια του μοντέλου αναγνώρισης

- Τα μοντέλα **shape\_predictor\_68\_face\_landmarks.dat** και **dlib\_face\_recognition\_resnet\_model\_v1.dat** έχουν εκπαιδευτεί σε συγκεκριμένα σύνολα δεδομένων.
  - Εάν τα πρόσωπα δεν ταιριάζουν με αυτά τα δεδομένα (π.χ., διαφορετικά χαρακτηριστικά, γωνίες, ηλικίες), η απόδοση μπορεί να είναι μειωμένη.
  - Η ακρίβεια μειώνεται επίσης αν τα πρόσωπα είναι μερικώς καλυμμένα (π.χ., με μάσκα, γυαλιά ή καπέλο).

### 5.4. Θέση και γωνία προσώπου

- Τα πρόσωπα που δεν είναι στραμμένα προς την κάμερα (π.χ., προφίλ) είναι πιο δύσκολο να αναγνωριστούν.
- Τα μοντέλα ανίχνευσης είναι πιο αποτελεσματικά για πρόσωπα που βλέπουν ευθεία την κάμερα.

## 5.5. Πολυπλοκότητα σκηνών

- Σε σκηνές με πολλά πρόσωπα ή σύνθετο φόντο, η απόδοση μπορεί να πέσει.
  - Ο εντοπισμός προσώπων μπορεί να γίνει πιο αργός, ειδικά αν χρησιμοποιούνται συσκευές με χαμηλή υπολογιστική ισχύ.

## 5.6. Επεξεργαστική Ικανότητα

- Το πρόγραμμα βασίζεται στη βιβλιοθήκη dlib, η οποία απαιτεί σημαντική επεξεργαστική ισχύ.
  - Σε συστήματα με χαμηλή CPU ή χωρίς GPU, η ταχύτητα εκτέλεσης μπορεί να είναι αργή, ειδικά για βίντεο υψηλής ανάλυσης.

## 5.7. Περιορισμοί στην επεξεργασία

- **Ασπρόμαυρη μετατροπή**
  - Η μετατροπή σε ασπρόμαυρη εικόνα μειώνει τον υπολογιστικό φόρτο, αλλά μπορεί να χάνονται πληροφορίες για την απόχρωση του δέρματος ή τον φωτισμό.
- **Αποκλίσεις χαρακτηριστικών**
  - Εάν το πρόσωπο έχει ιδιαίτερα χαρακτηριστικά ή οι εικόνες γνωστών προσώπων δεν είναι υψηλής ποιότητας, οι αποκλίσεις μπορεί να προκαλέσουν λάθος ταυτοποιήσεις.

# 6. Προτάσεις για βελτίωση

## 1. Χρήση εικόνων υψηλής ποιότητας

- Φρόντισε τα δεδομένα εκπαίδευσης (γνωστές εικόνες) να έχουν υψηλή ανάλυση και διαφορετικές γωνίες προσώπου.

## 2. Καλή φωτεινότητα και εστίαση

- Κατά τη λήψη βίντεο ή εικόνων, χρησιμοποίησε επαρκή φωτισμό και καλή εστίαση για καθαρά καρέ.

## 3. Αύξηση ανοχής (tolerance)

- Η συνάρτηση `face_recognition.compare_faces` έχει προεπιλεγμένη ανοχή 0.6. Μπορείς να αυξήσεις αυτή την τιμή αν θέλεις να μειώσεις τις πιθανότητες "μη ταύτισης".
- Για να γίνει η τροποποίηση του κάνε τα εξής βήματα:



- Βρες το σημείο :  
`matches = face_recognition.compare_faces(known_encodings, face_encoding)`
- Αν θέλεις να αυξήσεις την ανοχή, πρόσθεσε την παράμετρο `tolerance` στη συνάρτηση και όρισε μια μεγαλύτερη τιμή.  
`matches = face_recognition.compare_faces(known_encodings, face_encoding, tolerance=0.8)`
- Τι σημαίνει αυτό:
  - ✓ Μικρότερη τιμή (π.χ. 0.4): Πιο αυστηρή ταυτοποίηση, μειώνει τις πιθανότητες λανθασμένων ταυτοποιήσεων, αλλά μπορεί να "χάνει" πρόσωπα.
  - ✓ Μεγαλύτερη τιμή (π.χ. 0.8): Πιο ανεκτική ταυτοποίηση, αυξάνει τις πιθανότητες σωστών ταυτοποιήσεων, αλλά μπορεί να υπάρξουν ψευδείς θετικές.
- Αποθήκευσε το αρχείο Python αφού κάνεις την αλλαγή όπως αναφέρθηκε παραπάνω.

## 4. Βελτιστοποίηση απόδοσης

Αν χρησιμοποιείς βίντεο, περιορίσε την ανάλυση εισόδου ή μείωσε τα FPS (frames per second) για πιο γρήγορη εκτέλεση.

### Γιατί να περιορίσεις την ανάλυση εισόδου ή τα FPS;

- Υψηλή Ανάλυση: Όσο υψηλότερη είναι η ανάλυση του βίντεο, τόσο περισσότερα δεδομένα πρέπει να επεξεργαστεί το σύστημα. Αυτό αυξάνει το χρόνο επεξεργασίας ανά καρέ.
- Υψηλά FPS: Όταν το βίντεο έχει υψηλά καρέ ανά δευτερόλεπτο, το πρόγραμμα πρέπει να επεξεργάζεται περισσότερα καρέ ανά δευτερόλεπτο, γεγονός που επιβαρύνει την CPU ή την GPU.

### Πώς να περιορίσεις την ανάλυση εισόδου;

Μπορείς να μειώσεις την ανάλυση του βίντεο μέσω του OpenCV χρησιμοποιώντας τη μέθοδο `cv2.resize`. Αυτό θα μειώσει την ανάλυση του κάθε καρέ.

Παράδειγμα:

# Μείωση ανάλυσης βίντεο

`frame = cv2.resize(frame, (640, 480))` # Ορίζει την ανάλυση σε 640x480

### Πώς να περιορίσεις την ανάλυση εισόδου;

Μπορείς να μειώσεις την ανάλυση του βίντεο μέσω του OpenCV χρησιμοποιώντας τη μέθοδο `cv2.resize`. Αυτό θα μειώσει την ανάλυση του κάθε καρέ.

*Παράδειγμα:*

# Μείωση FPS

```
frame_count = 0
```

```
target_fps = 10 # Επεξεργασία μόνο 10 καρέ ανά δευτερόλεπτο
```

```
while video_capture.isOpened():
```

```
    ret, frame = video_capture.read()
```

```
    if not ret:
```

```
        break
```

```
    # Επεξεργασία μόνο εάν το καρέ είναι πολλαπλάσιο του (30/target_fps)
```

```
    if frame_count % (30 // target_fps) == 0:
```

```
        # Ανίχνευση προσώπων και επεξεργασία
```

```
        gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
        face_locations = face_recognition.face_locations(gray_frame)
```

```
        frame_count += 1
```

**Τι κάνει:**

- Αν το βίντεο είναι 30 FPS, επεξεργάζεται μόνο 10 καρέ το δευτερόλεπτο (1 καρέ κάθε 3).
- Βοηθά στην εξοικονόμηση πόρων, ειδικά σε συστήματα με χαμηλή επεξεργαστική ισχύ.

### **Συνδυασμός μείωσης ανάλυσης και FPS**

Ο συνδυασμός χαμηλότερης ανάλυσης και FPS μειώνει σημαντικά τον υπολογιστικό φόρτο.

*Παραδειγμα:*

```
frame_count = 0
```

```
target_fps = 10 # Επεξεργασία 10 καρέ ανά δευτερόλεπτο
```

```
while video_capture.isOpened():
```

```
    ret, frame = video_capture.read()
```

```
    if not ret:
```

```
        break
```

```

# Επεξεργασία μόνο κάθε (30/target_fps) καρέ
if frame_count % (30 // target_fps) == 0:

    # Μείωση ανάλυσης
    frame = cv2.resize(frame, (640, 480)) # Αλλαγή ανάλυσης σε 640x480

    # Ανίχνευση προσώπων
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    face_locations = face_recognition.face_locations(gray_frame)

    # Σχεδιασμός πλαισίων και επεξεργασία
    for top, right, bottom, left in face_locations:
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.imshow("Video", frame)

frame_count += 1

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

video_capture.release()
cv2.destroyAllWindows()

```

### Οφέλη της Βελτιστοποίησης

- **Χαμηλότερος Υπολογιστικός Φόρτος:** Το πρόγραμμα θα τρέχει πιο γρήγορα και θα καταναλώνει λιγότερους πόρους.
- **Σταθερή Απόδοση:** Το πρόγραμμα θα είναι πιο ανθεκτικό σε βίντεο με υψηλή ανάλυση ή μεγάλα FPS.
- **Ευκολότερη Δοκιμή και Ρύθμιση:** Μπορείς να δοκιμάσεις το πρόγραμμα σε διαφορετικά περιβάλλοντα χωρίς να ανησυχείς για την ισχύ του συστήματος.

### Νέος κώδικας προσαρμοσμένος

```

import cv2

import face_recognition

import os

```

```

# Έλεγχος για τα αρχεία .dat

MODEL_PATH = "models"

SHAPE_PREDICTOR_PATH=os.path.join(MODEL_PATH,
"shape_predictor_68_face_landmarks.dat")

FACE_RECOGNITION_MODEL_PATH=os.path.join(MODEL_PATH,
"dlib_face_recognition_resnet_model_v1.dat")

If      not      os.path.exists(SHAPE_PREDICTOR_PATH)      or      not
os.path.exists(FACE_RECOGNITION_MODEL_PATH):

    print("Τα απαραίτητα αρχεία .dat δεν βρέθηκαν. Ελέγξτε τον φάκελο
models.")
    exit()
print("Τα αρχεία μοντέλων βρέθηκαν επιτυχώς!")

# Φόρτωση γνωστών προσώπων

KNOWN_FACES_PATH = "known_faces"

if not os.path.exists(KNOWN_FACES_PATH):

    print("Ο φάκελος γνωστών προσώπων δεν βρέθηκε. Δημιουργήστε τον και
προσθέστε φακέλους με εικόνες.")

    exit()

known_encodings = []

known_names = []

# Φόρτωση γνωστών προσώπων από τον φάκελο

for person_name in os.listdir(KNOWN_FACES_PATH):

    person_folder = os.path.join(KNOWN_FACES_PATH, person_name)

    if not os.path.isdir(person_folder):

        continue

    for image_name in os.listdir(person_folder):

        image_path = os.path.join(person_folder, image_name)

        image = face_recognition.load_image_file(image_path)

        encodings = face_recognition.face_encodings(image)

```

```

        if len(encodings) > 0:

            known_encodings.append(encodings[0])

            known_names.append(person_name)

    print(f'Φορτώθηκαν {len(known_encodings)} γνωστά πρόσωπα.')

    # Επιλογή εισόδου

    choice = input("Πατήστε 1 για κάμερα ή 2 για αποθηκευμένο βίντεο: ")

    if choice == "1":

        video_source = 0 # Κάμερα

    elif choice == "2":

        video_source = input("Εισάγετε το πλήρες μονοπάτι του βίντεο: ").strip()

        if not os.path.exists(video_source):

            print(f"Το αρχείο βίντεο δεν βρέθηκε στο μονοπάτι: {video_source}")

            exit()

    else:

        print("Μη έγκυρη επιλογή.")

        exit()

    # Ρυθμίσεις για ανάλυση

    target_resolution = input("Επιλέξτε ανάλυση (π.χ. 640x480 ή αφήστε κενό για προεπιλογή): ").strip()

    if target_resolution:

        try:

            width, height = map(int, target_resolution.split("x"))

        except ValueError:

            print("Μη έγκυρη ανάλυση. Χρήση προεπιλογής (640x480).")

            width, height = 640, 480

    else:

```

```

width, height = 640, 480

# Ρυθμίσεις ανοχής (tolerance)

try:

    tolerance = float(input("Εισάγετε την ανοχή ταύτισης (προεπιλογή: 0.6):
").strip() or 0.6)

except ValueError:

    print("Μη έγκυρη τιμή. Χρήση προεπιλεγμένης ανοχής: 0.6.")

    tolerance = 0.6

# Έναρξη βίντεο

video_capture = cv2.VideoCapture(video_source)

while video_capture.isOpened():

    ret, frame = video_capture.read()

    if not ret:

        print("Δεν είναι δυνατή η ανάγνωση από την πηγή.")

        break

    # Μείωση ανάλυσης καρέ

    frame = cv2.resize(frame, (width, height))

    # Μετατροπή του καρέ σε ασπρόμαυρο

    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Ανίχνευση προσώπων

    face_locations = face_recognition.face_locations(gray_frame)

    face_encodings = face_recognition.face_encodings(frame, face_locations)

    # Μετατροπή του ασπρόμαυρου καρέ σε έγχρωμο για σχεδιασμό κόκκινων
    πλαισίων

    color_frame = cv2.cvtColor(gray_frame, cv2.COLOR_GRAY2BGR)

    # Σχεδίαση πλαισίων και ονομάτων γύρω από τα πρόσωπα

```

```

for face_location, face_encoding in zip(face_locations, face_encodings):

    matches = face_recognition.compare_faces(known_encodings,
face_encoding, tolerance)

    name = "Άγνωστος"

    if True in matches:

        match_index = matches.index(True)

        name = known_names[match_index]

    top, right, bottom, left = face_location

    cv2.rectangle(color_frame, (left, top), (right, bottom), (0, 0, 255), 2)

    cv2.putText(color_frame, name, (left, top - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2)

# Προβολή βίντεο

cv2.imshow("Αναγνώριση Προσώπων", color_frame)

# Έξοδος με 'q'

if cv2.waitKey(1) & 0xFF == ord('q'):

    break

# Καθαρισμός

video_capture.release()

cv2.destroyAllWindows()

```

## 7. Συμπεράσματα

Η εργασία που ολοκληρώθηκε αφορά την ανάπτυξη ενός συστήματος ανίχνευσης και αναγνώρισης προσώπων χρησιμοποιώντας Python και σχετικές βιβλιοθήκες όπως dlib, face\_recognition, και OpenCV. Μέσα από τη διαδικασία εγκατάστασης, ελέγχου και υλοποίησης του κώδικα, προέκυψαν σημαντικά σημεία που πρέπει να ληφθούν υπόψη για την επιτυχή εκτέλεση και τη βελτιστοποίηση της απόδοσης.

### 7.1. Βασικά σημεία:

#### 1. Εγκατάσταση και Προαπαιτούμενα:

- Η Python αποτελεί τη βάση της υλοποίησης, και η σωστή εγκατάστασή της μαζί με τα εργαλεία pip και cmake ήταν κρίσιμη.
- Οι βιβλιοθήκες dlib, face\_recognition, opencv-python και scipy εγκαταστάθηκαν και ρυθμίστηκαν, επιτρέποντας τη λειτουργία του συστήματος.
- **Λειτουργικότητα του Συστήματος:**
- Το πρόγραμμα υποστηρίζει ανίχνευση και αναγνώριση προσώπων είτε από ζωντανή κάμερα είτε από αποθηκευμένα βίντεο.
- Τα δεδομένα των γνωστών προσώπων αποθηκεύονται σε δομημένους φακέλους, γεγονός που επιτρέπει εύκολη διαχείριση και επέκταση.

#### 2. Τεχνικές Βελτιστοποίησης:

- Η μετατροπή των καρέ σε ασπρόμαυρο (grayscale) μειώνει τον υπολογιστικό φόρτο και αυξάνει την ταχύτητα επεξεργασίας, ενώ παράλληλα διατηρεί την ακρίβεια ανίχνευσης.
- Η παράμετρος "tolerance" στη συνάρτηση compare\_faces προσαρμόζεται για να βελτιώσει την ακρίβεια της αναγνώρισης.

#### 3. Προβλήματα και Λύσεις:

- Οι δυσκολίες με τη διαχείριση του PATH και των εγκαταστάσεων διορθώθηκαν μέσω ενδεδειγμένων ελέγχων στις μεταβλητές συστήματος.
- Η σωστή παραμετροποίηση των βιβλιοθηκών και των μοντέλων .dat διασφάλισε την πλήρη λειτουργικότητα.



#### 4. Περιορισμοί και Προτάσεις:

- ο Το σύστημα επηρεάζεται από εξωτερικούς παράγοντες όπως ο θόρυβος, η φωτεινότητα, και η ποιότητα των εικόνων/βίντεο.
- ο Προτείνεται η χρήση υψηλής ανάλυσης εικόνων για τα "γνωστά πρόσωπα" και η εφαρμογή τεχνικών προεπεξεργασίας για τη βελτίωση της απόδοσης σε περιβάλλοντα με χαμηλό φωτισμό.

#### 5. Επεκτασιμότητα και Εφαρμογές:

- ο Το πρόγραμμα μπορεί να προσαρμοστεί σε διάφορες εφαρμογές όπως συστήματα ασφαλείας, διαχείριση εισόδου, ή κοινωνικές εφαρμογές.
- ο Η προσθήκη υποστήριξης GPU στο μέλλον θα μπορούσε να βελτιώσει ακόμα περισσότερο την ταχύτητα ανίχνευσης.

## 7..2. Τελική Εκτίμηση

Η εργασία αποτελεί μια εξαιρετική εισαγωγή στη χρήση εργαλείων τεχνητής νοημοσύνης για αναγνώριση προσώπων. Παρά τα εμπόδια που προέκυψαν επιτεύχθηκε ένα λειτουργικό και αποδοτικό σύστημα. Αυτό το project αποτελεί τη βάση για πιο σύνθετες εφαρμογές αναγνώρισης προσώπων, συνδυάζοντας την ακρίβεια με τη φιλικότητα προς τον χρήστη.

## 7..3. Πιο Σύνθετες Εφαρμογές Αναγνώρισης Προσώπων

Η τεχνολογία που αναπτύχθηκε στο παραπάνω project μπορεί να επεκταθεί και να εφαρμοστεί σε διάφορους τομείς με πιο σύνθετες λειτουργίες. Ακολουθούν μερικά παραδείγματα:

### 1. Συστήματα Ασφαλείας και Έλεγχος Πρόσβασης

- **Περιγραφή:** Χρήση αναγνώρισης προσώπου για τον έλεγχο πρόσβασης σε κτίρια, γραφεία ή ευαίσθητες περιοχές.
- **Εφαρμογές:**
  - ο Έλεγχος εισόδου σε γραφεία μόνο για εξουσιοδοτημένους χρήστες.
  - ο Αυτόματη ενεργοποίηση συναγερμού εάν εντοπιστεί άγνωστο πρόσωπο σε συγκεκριμένες ζώνες.
- **Βελτιώσεις:**
  - ο Υλοποίηση βάσης δεδομένων με επίπεδα πρόσβασης για διαφορετικούς χρήστες.
  - ο Χρήση κάμερας νυχτερινής όρασης για αναγνώριση σε χαμηλό φωτισμό.

## **2. Προσωποποιημένες Εμπειρίες Πελατών**

- **Περιγραφή:** Εφαρμογή αναγνώρισης προσώπου σε εμπορικά καταστήματα για την παροχή προσωποποιημένων υπηρεσιών.
- **Εφαρμογές:**
  - Αυτόματη αναγνώριση τακτικών πελατών και παροχή προσφορών ή εκπτώσεων.
  - Προσωποποιημένες διαφημίσεις σε οθόνες ανάλογα με την ηλικία ή το φύλο του πελάτη.
- **Βελτιώσεις:**
  - Ανάλυση συναισθημάτων για καλύτερη προσαρμογή των διαφημίσεων.

## **3. Σύστημα Παρουσιολογίου σε Σχολεία και Εταιρείες**

- **Περιγραφή:** Χρήση αναγνώρισης προσώπου για την αυτόματη καταγραφή παρουσιών.
- **Εφαρμογές:**
  - Αυτόματη αναγνώριση των μαθητών/εργαζομένων κατά την είσοδό τους στην αίθουσα.
  - Αποστολή ειδοποίησης σε περίπτωση απουσίας.
- **Βελτιώσεις:**
  - Σύνδεση με το σύστημα διαχείρισης μαθημάτων ή το HR software για καταγραφή δεδομένων.

## **4. Εφαρμογές Υγείας**

- **Περιγραφή:** Χρήση αναγνώρισης προσώπου για τη βελτίωση της υγείας και της φροντίδας ασθενών.
- **Εφαρμογές:**
  - Αναγνώριση ασθενών σε νοσοκομεία για ταχύτερη εξυπηρέτηση.
  - Παρακολούθηση συναισθημάτων ασθενών (άγχος, θλίψη) σε ψυχολογικές κλινικές.
- **Βελτιώσεις:**
  - Συνδυασμός με άλλους αισθητήρες (π.χ. μέτρηση θερμοκρασίας ή καρδιακών παλμών).

## **5. Επιτήρηση Κοινού και Εντοπισμός Υπόπτων**

- **Περιγραφή:** Αναγνώριση ατόμων σε δημόσιους χώρους για την πρόληψη εγκληματικών ενεργειών.
- **Εφαρμογές:**
  - Εντοπισμός υπόπτων από λίστα καταζητούμενων.
  - Παρακολούθηση της κίνησης σε πολυσύχναστους χώρους (π.χ. αεροδρόμια, γήπεδα).

## **6. Ανάλυση Συναθροίσεων**

- **Περιγραφή:** Παρακολούθηση της συμπεριφοράς και της κίνησης ατόμων σε δημόσιους χώρους ή εκδηλώσεις.
- **Εφαρμογές:**
  - Υπολογισμός πλήθους για τη διαχείριση εισόδου/εξόδου.
  - Ανίχνευση ατόμων σε κατάσταση κινδύνου (π.χ. κατά τη διάρκεια φυσικών καταστροφών).
- **Βελτιώσεις:**
  - Προσθήκη ανάλυσης συναισθημάτων ή εντοπισμού ατυχημάτων.
- **Βελτιώσεις:**
  - Σύνδεση με κεντρικές βάσεις δεδομένων για real-time ταυτοποίηση.

### **Γενικές Βελτιώσεις για Σύνθετες Εφαρμογές:**

- **Υποστήριξη GPU:** Χρήση υπολογιστικής ισχύος για γρηγορότερη ανάλυση μεγάλου όγκου δεδομένων.
- **Σύνδεση με Cloud:** Αποθήκευση και επεξεργασία δεδομένων σε cloud για καλύτερη επεκτασιμότητα.
- **Επεξεργασία Μεγάλων Βάσεων Δεδομένων:** Εφαρμογή αλγορίθμων για ταχύτερη αναζήτηση και ταυτοποίηση.

Αυτές οι εφαρμογές αποτελούν το επόμενο επίπεδο χρήσης της τεχνολογίας αναγνώρισης προσώπου και ανοίγουν τον δρόμο για καινοτόμες λύσεις σε διάφορους τομείς.