

MAI4CAREU

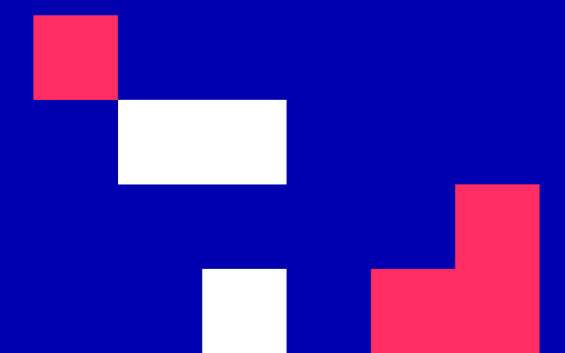
Master programmes in Artificial
Intelligence 4 Careers in Europe

University of Cyprus

HUMAN-CENTERED INTELLIGENT USER INTERFACES - MAI648

Marios Belk

2022



LAB 8

Face-based User Identification

CONTENTS

- Laboratory Description
 - TRUSTID Project
 - Face-based Identification
 - About Python Face Recognition
 - License
- Requirements
 - Installation
 - Main Features of the Face Recognition Library
 - Example Codes

LAB 8

Laboratory Description

- In this laboratory you will familiarize yourself with using state-of-the-art libraries for face-based user identification
- Recap on the Erasmus+ TRUSTID project
- Then we will follow the steps as documented in the following Face Recognition system - https://github.com/ageitgey/face_recognition

LAB 8

Face-based User Identification

- Image/video based Biometric system (face recognition)
- **Goal:** Identify users from pre-recorded image/video dataset
- System constrains:
 - Noisy image data (consumer grade webcams, acquisition issues, unconstrained environments, ...)
 - Accurate and reliable system (requirement).
 - Computational performance concerns.

Source: Institute of Systems and Robotics, University of Coimbra

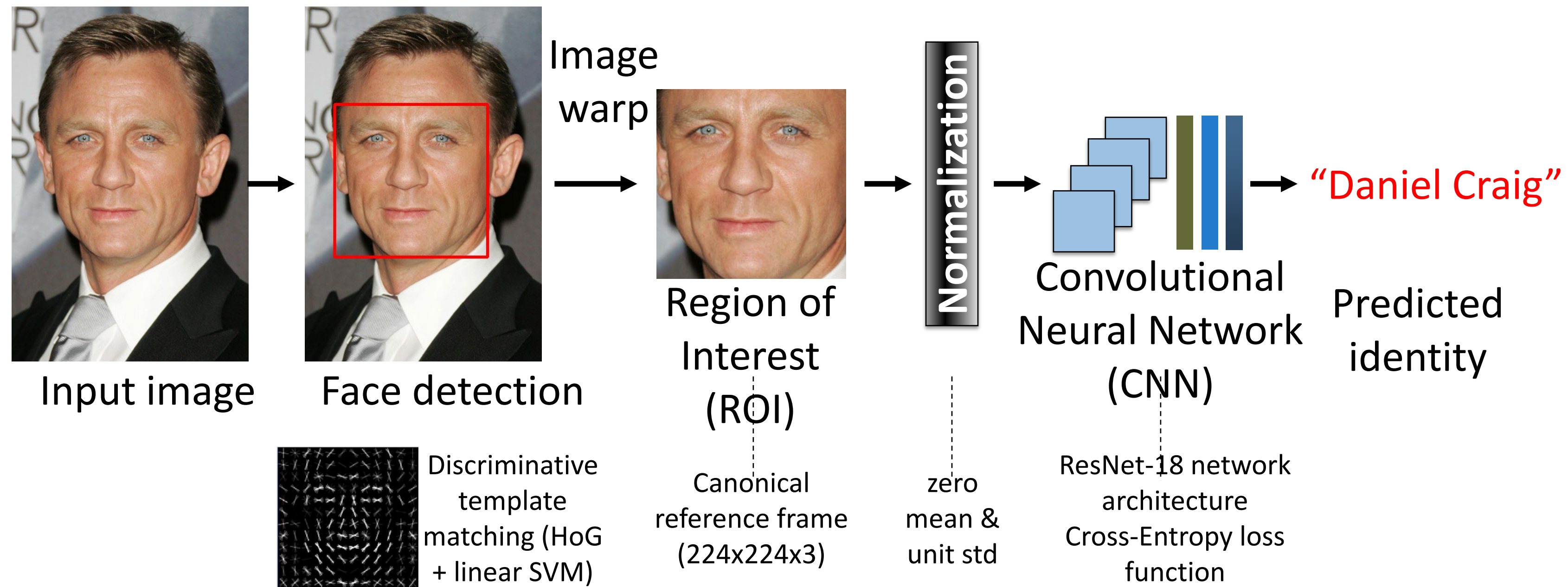
LAB 8

Face-based User Identification

- Training stage (offline):
 - Assemble user's image database.
 - Face detection (locate faces in all images)
 - Data augmentation (add "virtual" variation to database images, p.e. geometric and color transformations).
 - Learn multiclass classifier from corresponding image/users examples.
- Testing stage (online):
 - Face detection
 - Predict user identity using the pretrained classifier model

Source: Institute of Systems and Robotics, University of Coimbra

Face Recognition System Overview



Source: Institute of Systems and Robotics, University of Coimbra

About Python Face Recognition

- Face recognition aims to recognize faces from Python or from the command line with a simple face recognition library
- The library is built using dlib's state-of-the-art face recognition built with deep learning. According to the authors' testing, the model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

LAB 8

License

- ageitgey/face_recognition is licensed under the
- **MIT License**
- A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

LAB 8

Requirements

- Python 3.3+ or Python 2.7
- macOS or Linux or Windows

LAB 8

Installation

- Git: Source control versioning system
- Download Git for Windows in order to be able to run git commands in a Windows terminal
- Download link: <https://git-scm.com/download/win>
- pip3 install dlib
- pip3 install face_recognition
- pip3 install opencv-python
- https://github.com/ageitgey/face_recognition/issues/175#issue-257710508

LAB 8

Main Features of the Face Recognition Library

- Find faces in pictures
- Find and manipulate facial features in pictures
- Identify faces in pictures

LAB 8

Find faces in pictures

```
import face_recognition

image = face_recognition.load_image_file("your_file.jpg")
face_locations = face_recognition.face_locations(image)
```


LAB 8

Find and manipulate facial features in pictures

```
import face_recognition

image = face_recognition.load_image_file("your_file.jpg")
face_landmarks_list = face_recognition.face_landmarks(image)
```

LAB 8

Identify faces in pictures

```
import face_recognition

known_image = face_recognition.load_image_file("biden.jpg")
unknown_image = face_recognition.load_image_file("unknown.jpg")
biden_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[0]
results = face_recognition.compare_faces([biden_encoding], unknown_encoding)
```

LAB 8

Recognize faces in images and identify who they are

```
import face_recognition

picture_of_me = face_recognition.load_image_file("me.png")
my_face_encoding = face_recognition.face_encodings(picture_of_me) [0]

unknown_picture = face_recognition.load_image_file("unknown.png")
unknown_face_encoding = face_recognition.face_encodings(unknown_picture) [0]
```

LAB 8

Recognize faces in images and identify who they are

```
results = face_recognition.compare_faces([my_face_encoding],  
unknown_face_encoding)  
  
if results[0] == True:  
    print("It's a picture of me!")  
else:  
    print("It's not a picture of me!")
```


LAB 8

Identify specific facial features in a photograph

```
from PIL import Image, ImageDraw
import face_recognition

# Load the jpg file into a numpy array
image = face_recognition.load_image_file("two_people.jpg")

# Find all facial features in all the faces in the image
face_landmarks_list = face_recognition.face_landmarks(image)

print("I found {} face(s) in this
photograph.".format(len(face_landmarks_list)))
```

LAB 8

Identify specific facial features in a photograph

```
pil_image = Image.fromarray(image)
d = ImageDraw.Draw(pil_image)

for face_landmarks in face_landmarks_list:

    # Print the location of each facial feature in this image
    for facial_feature in face_landmarks.keys():
        print("The {} in this face has the following points:
        {}".format(facial_feature, face_landmarks[facial_feature]))
```

LAB 8

Identify specific facial features in a photograph

```
# Let's trace out each facial feature in the image with a line!  
for facial_feature in face_landmarks.keys():  
    d.line(face_landmarks[facial_feature], width=5)  
  
# Show the picture  
pil_image.show()
```

LAB 8

Sources

- Erasmus+ TRUSTID Project – <https://trustid-project.eu>
- Face-based Identification by Institute of Systems and Robotics, University of Coimbra
- https://github.com/ageitgey/face_recognition

MAI4CAREU

Master programmes in Artificial
Intelligence 4 Careers in Europe

Thank you.