



Facial Recognition Login

Login por reconhecimento facial

Protótipo da solução

Trabalho Laboratorial

Engenharia de Sistemas e Serviços

EI 2023/24

Grupo 5 PL1

Francisco Ferreira Bento nº2201788

Pedro Diogo Nunes de Carvalho nº2192007

Leiria, Junho de 2024



Inicio Projeto: Março de 2024

Fim do Projeto: Junho de 2024

Resumo

A tecnologia de reconhecimento facial ganhou destaque nos últimos anos devido às suas diversas aplicações em variados campos. As webcams desempenham um papel fundamental no reconhecimento facial em dispositivos como smartphones, computadores e outros equipamentos.

O reconhecimento facial é uma tecnologia biométrica que identifica e autentica os utilizadores, com base nas suas características faciais, utiliza algoritmos complexos para analisar e comparar características faciais únicas, como a distância entre os olhos, formato do nariz e contornos do rosto. Este projeto tem como objetivo oferecer um método de login intuitivo, rápido e eficaz, dos utilizadores através de identificação facial.

É uma solução “quality of life”, dando ao utilizador uma chave universal para fazer login em diversos websites.

Os alunos propõem o desenvolvimento de uma aplicação de reconhecimento facial para login em websites, utilizando a webcam do computador, que seria instalada como uma extensão do navegador de internet, comunicando com a base de dados alojada na cloud, fornecendo assim um serviço a qualquer hora e em qualquer lugar.

Este projeto faz parte de unidade curricular de terceiro ano, Engenharia de Sistemas e Serviços, do curso de Engenharia Informática do Instituto Politécnico de Leiria, a decorrer no segundo semestre.

Palavras Chave: “Tecnologia”; “Reconhecimento Facial”; “Automação de login”

Abstract

Facial recognition technology has gained prominence in recent years due to its many applications in various fields. Webcams play a key role in facial recognition on devices such as smartphones, computers and other equipment.

Facial recognition is a biometric technology that identifies and authenticates users based on their facial features, using complex algorithms to analyze and compare unique facial features such as the distance between the eyes, nose shape and contours of the face.

This project aims to offer an intuitive, fast and effective method of logging users in via facial identification.

It is a "quality of life" solution, giving the user a universal key to log in to various websites.

Students propose the development of facial recognition application for logging into websites, using the computer's webcam, which would be installed as an extension of the internet browser, communicating with the database hosted in the cloud, thus providing a service anytime, anywhere.

This project is part of the Systems and Services Engineering curricular unit of the Computer Engineering course at the Polytechnic Institute of Leiria, taking place in the second semester of the third year of the course.

Keywords: “Technology”; “Facial Recognition”; “Login Automation”

Índice

Resumo	3
Abstract.....	4
Lista de Figuras.....	5
Introdução.....	6
Objetivos do trabalho	6
Descrição do Cenário de Implementação	7
Descrição do Cenário de teste	8
Objetivos dos testes e resultados	9
Funcionamento do Cenário e Resultado dos testes	9
Manual de Instalação e configuração	15
Conclusões	16
Bibliografia	17

Lista de Figuras

Figure 1- Esquema do cenário de implementação	7
Figure 2 -Legenda dos itens do cenário	7
Figure 3 -Esquema do cenário de implementado.....	8
Figure 4- Cenário de testes real.	8
Figure 5 -Vista das configurações para os scripts Python.....	9
Figure 6 -Vista da pasta das fotos para usar no script Python.....	9
Figure 7 -Vista da deteção do utilizador	10
Figure 8-Vista da deteteção de um utilizador e outro não reconhecido	10
Figure 9 - Vista da webcam a identificar os dois utilizadores	11
Figure 10 - Vista do Script em execução no computador Receiver	12
Figure 11 - Vista do Receiver se não receber dados válidos.....	13
Figure 12- Vista da Receção dos dados de login no Receiver	13
Figure 13- Vista do código Python onde esta inserido o website.....	13

Introdução

A evolução das soluções biométricas permitem aos utilizadores, de forma fiável, serem reconhecidos pelo sistema e desbloquear o acesso aos meios onde pretendem aceder, sejam físicos ou virtuais.

A utilização desta tecnologia nos sistemas torna-se fundamental para alcançar níveis superiores de segurança, utiliza características físicas únicas e intransmissíveis proporcionando um nível de autenticação infalível. Estas características individuais permitem a facilidade nos acessos pelas pessoas autorizadas e impedem riscos de falsificação ou intrusões não desejadas.

As webcams são fundamentais para capturar imagens faciais ou fluxos de vídeo usados em sistemas de reconhecimento facial. A qualidade e os recursos das webcams influenciam diretamente a precisão e a fiabilidade do reconhecimento facial.

Objetivos do trabalho

A solução apresentada neste projeto, propõe o desenvolvimento de um aplicativo que agiliza o processo de autenticação dos utilizadores, eliminando a necessidade de inserção manual de credenciais durante o login, considerando a crescente demanda por logins em plataformas de internet. Ao aplicar o reconhecimento facial como único método de autenticação, o processo torna-se significativamente mais rápido e potencialmente mais seguro. Esta proposta inspira-se em três projetos existentes: Windows Hello, Apple Face ID e sistemas terminais de reconhecimento facial de entradas e saídas.

Descrição do Cenário de Implementação

No cenário de implementação vamos utilizar dois computadores ligados em rede, com um Router/Switch Mikrotik para simular uma cloud. O computador1 é a máquina que vai correr o aplicativo "Sender". O Computador2 é a máquina que vai correr o aplicativo "Receiver".

Vamos utilizar uma aplicação em Python, cujo objetivo será ligar a webcam do pc, identificar a face que está a ser examinada e enviar as credenciais correspondentes.

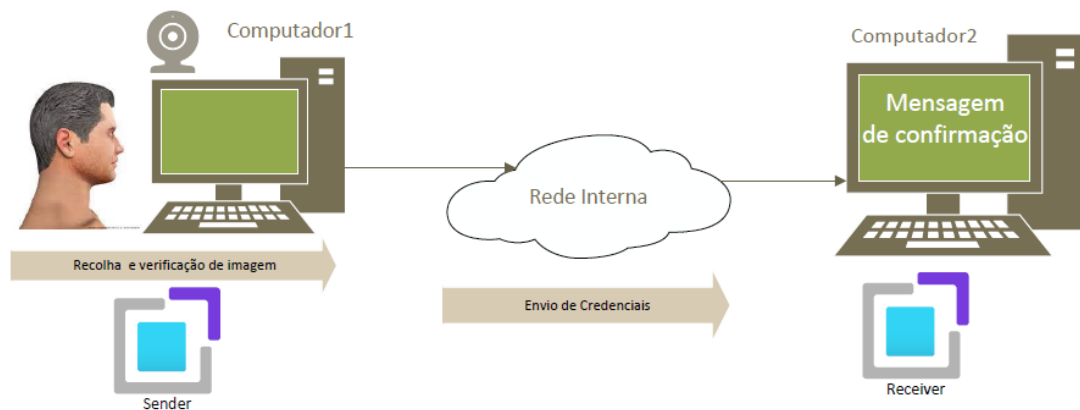


Figure 1- Esquema do cenário de implementação



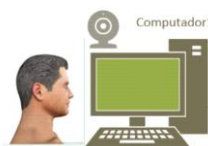


Equipamento	Descrição
 Sender	Aplicação em python, cujo objetivo será ligar a webcam do pc, identificar a face que está a ser examinada e enviar as credenciais correspondentes.
 Receiver	Aplicação em python, cujo objetivo será receber as credenciais que o "Sender" enviará.
 Computador1	O Computador1 é a máquina que vai correr o aplicativo "Sender"
 Computador2	O Computador2 é a máquina que vai correr o aplicativo "Receiver"
 Rede Interna	Para a rede interna irá ser usado um router/switch mikrotik, para fazer a ligação do computador1 com o computador2

Figure 2 -Legenda dos itens do cenário

Descrição do Cenário de teste

Para o desenvolvimento do protótipo, optou-se pela linguagem de programação Python, com código configurado para funcionar em dois computadores distintos.

No primeiro computador, executa-se um script Python que estabelece um canal de comunicação com o segundo computador, o qual se encontra em modo de espera. No primeiro computador existe duas pastas, uma com fotos dos utilizadores, outra com as suas credenciais de acesso. Este script também inicia a interface de utilizador, ativa a webcam, capta e identifica a face do utilizador, e transmite os dados de login para o segundo computador. Posteriormente, no segundo computador, o website é automaticamente aberto, os campos de login são preenchidos automaticamente e a sessão é iniciada.

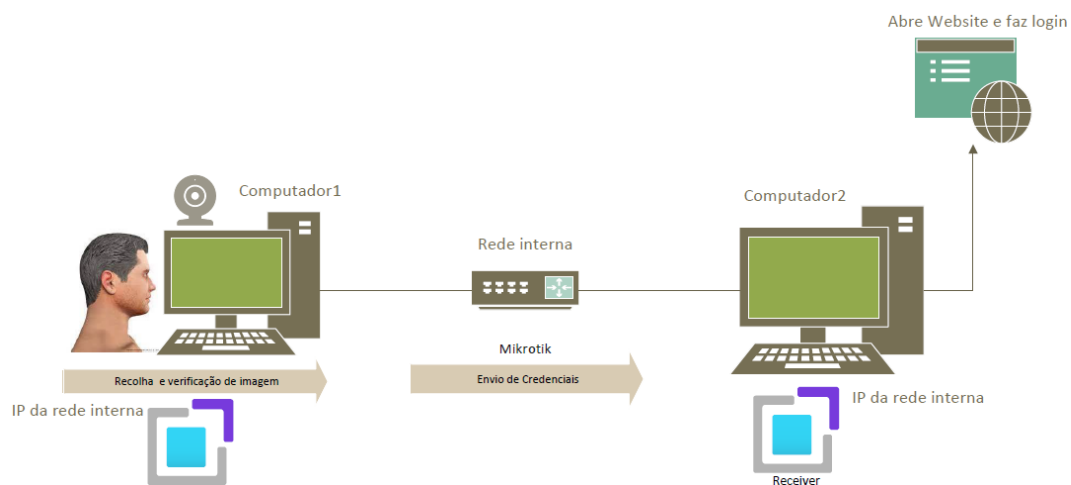


Figure 3 -Esquema do cenário de implementado

O cenário real usado para realizar testes foi configurado com um router Mikrotik, no qual está configurada uma rede privada nas portas ethernet. Dois computadores foram ligados nessa rede privada para simular a ligação entre um PC e um Servidor.

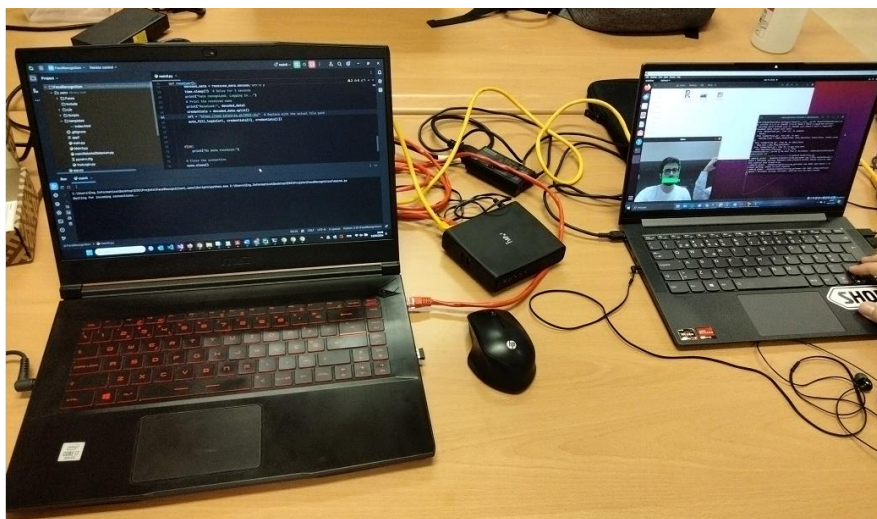


Figure 4- Cenário de testes real.

Objetivos dos testes e resultados

O principal objetivo deste protótipo é permitir que o utilizador aceda a um website utilizando a sua face como método de autenticação. Atualmente, o protótipo está configurado para efetuar login apenas no site na plataforma de Ensino à Distância do IPL, mas é possível adicionar mais faces de utilizadores para autenticação. Durante os testes, conseguimos identificar a face de cada utilizador e as respetivas credenciais associadas à face detetada pela aplicação. Após a identificação, o website é aberto automaticamente, os campos de login são preenchidos automaticamente e o acesso à plataforma da escola é concedido.

Funcionamento do Cenário e Resultado dos testes

- No computador que executa o script do Sender, as credenciais dos utilizadores estão armazenadas em texto simples, o que representa uma falha de segurança significativa. Tem de ser criados ficheiros TXT com o nome do utilizador igual ao nome da foto. Este ficheiro TXT tem o utilizadores e password.

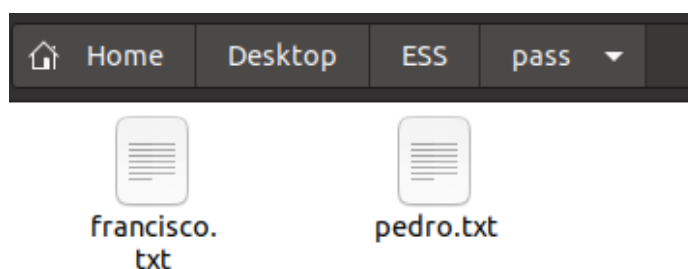


Figure 5 -Vista das configurações para os scripts Python

- Na pasta das fotos estão inseridas as fotos dos utilizadores que pretendemos que sejam identificados pela aplicação. A camara vai comparar a imagem real com a foto presente nesta pasta.



Figure 6 -Vista da pasta das fotos para usar no script Python

- Ao executar o script no computador Sender, abre a Webcam e inicia a comparação da face real com a foto presente na pasta. Como se pode verificar na imagem abaixo, a webcam deteta o nome do utilizador que esta a ser detetado, este nome corresponde ao nome da foto armazenada na pasta.

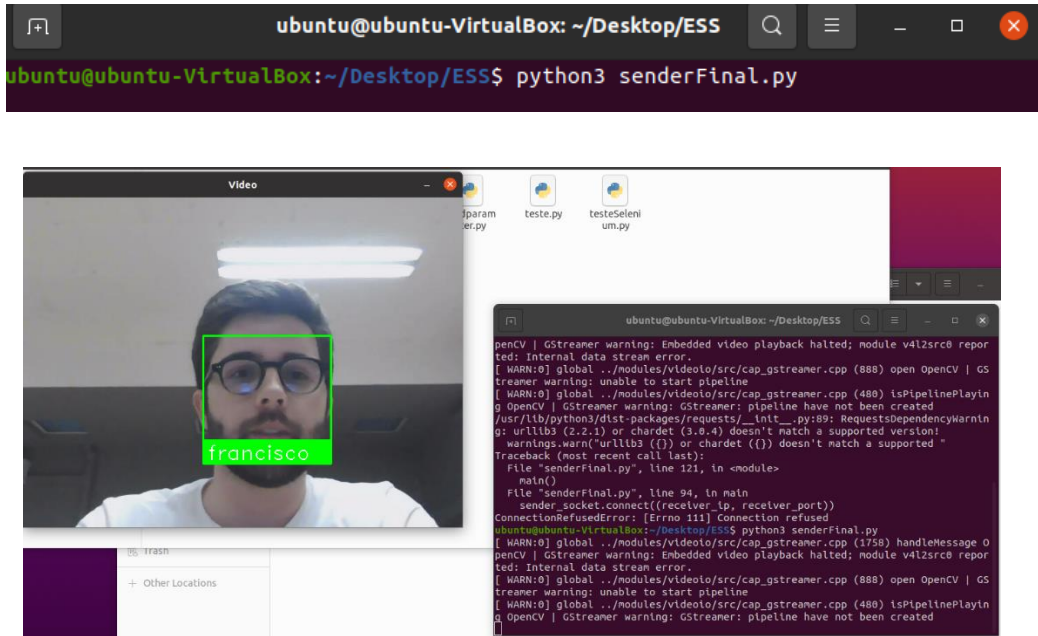


Figure 7 -Vista da deteção do utilizador

- Durante o processo de reconhecimento ao colocar outro utilizador em frente á Webcam, ele apenas deteta o utilizador que reconhece, identifica-o pelo nome correspondente é foto armazenada na pasta. O outro utilizador aparece com identificação “Desconhecido”.

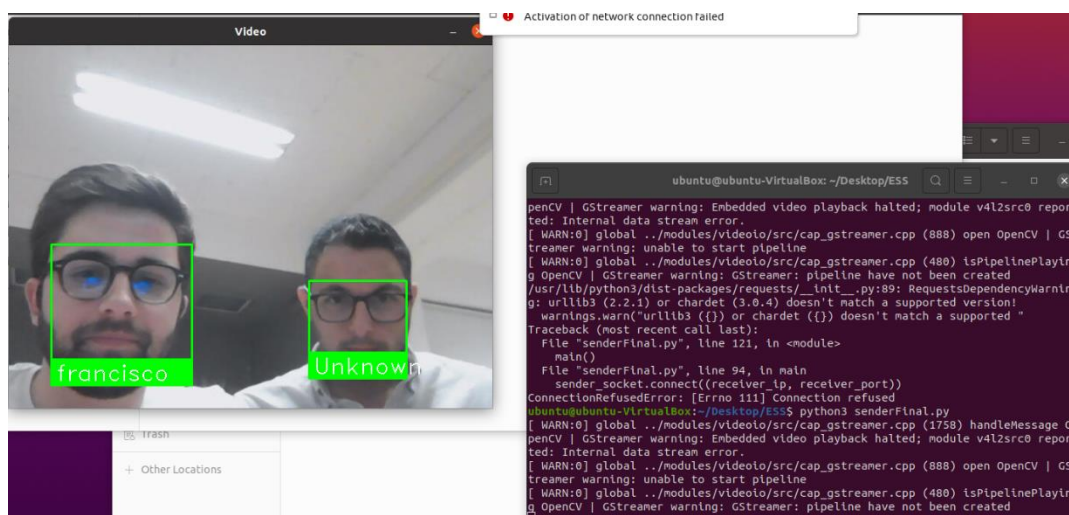


Figure 8-Vista da deteção de um utilizador e outro não reconhecido

- De seguida foi realizado novo teste já com a foto dos dois utilizadores armazenada na pasta do computador Sender, e como se pode ver na imagem abaixo, a webcam mostra os dois utilizadores identificados pelo mesmo nome que corresponde á sua foto.

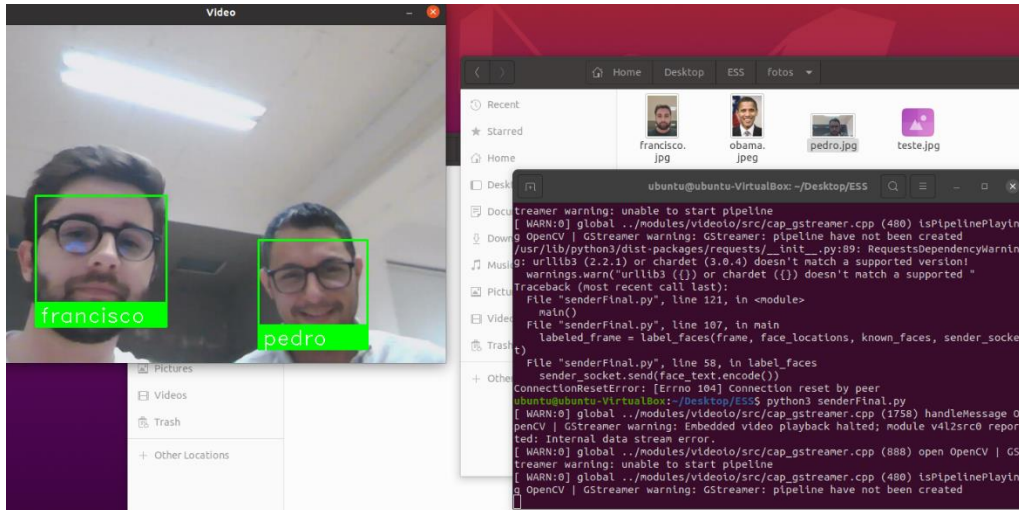
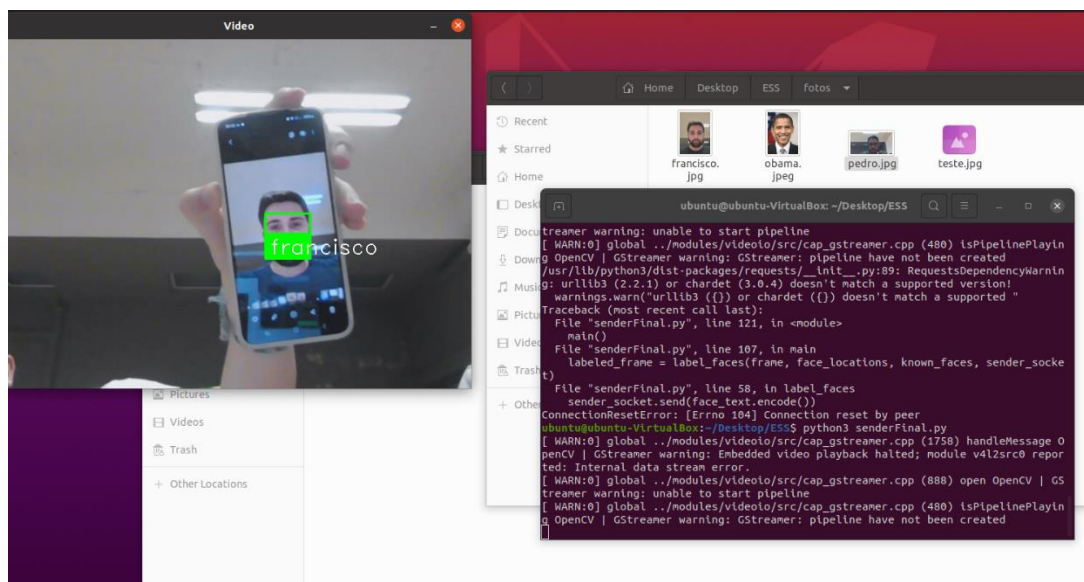
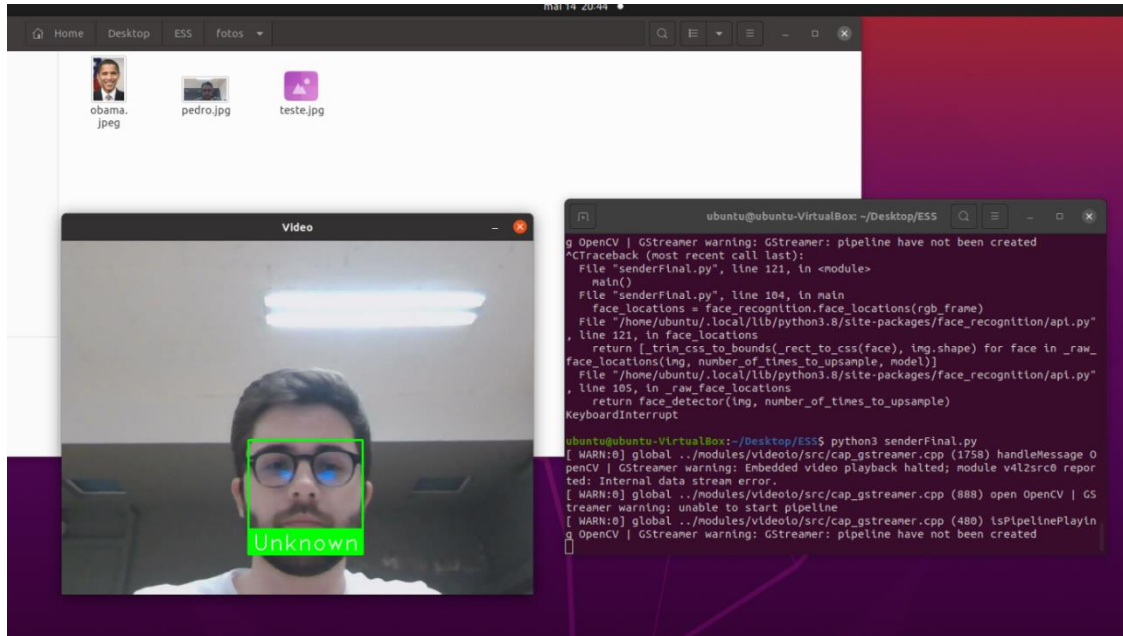


Figure 9 - Vista da webcam a identificar os dois utilizadores

- Outro teste realizado foi colocar um smartphone em frente á webcam do computador Sender, e verifica-se que a webcam não consegue distinguir entre uma foto e o rosto real. Identificamos aqui uma falha de segurança que pode pôr em causa o funcionamento da aplicação e a segurança da informação dos utilizadores. Esta falha teria de ser analisada e resolvida.



- Ao remover-mos a foto do utilizador, da pasta onde são armazenadas, verifica-se que a webcam lê a face, mas não consegue autenticar o utilizador, e assim não procede à face seguinte, que é o envio de dados (login) do utilizador para o Receiver.



- No computador Receiver é executado o script, que fica á espera de receber comunicação do computador Sender.

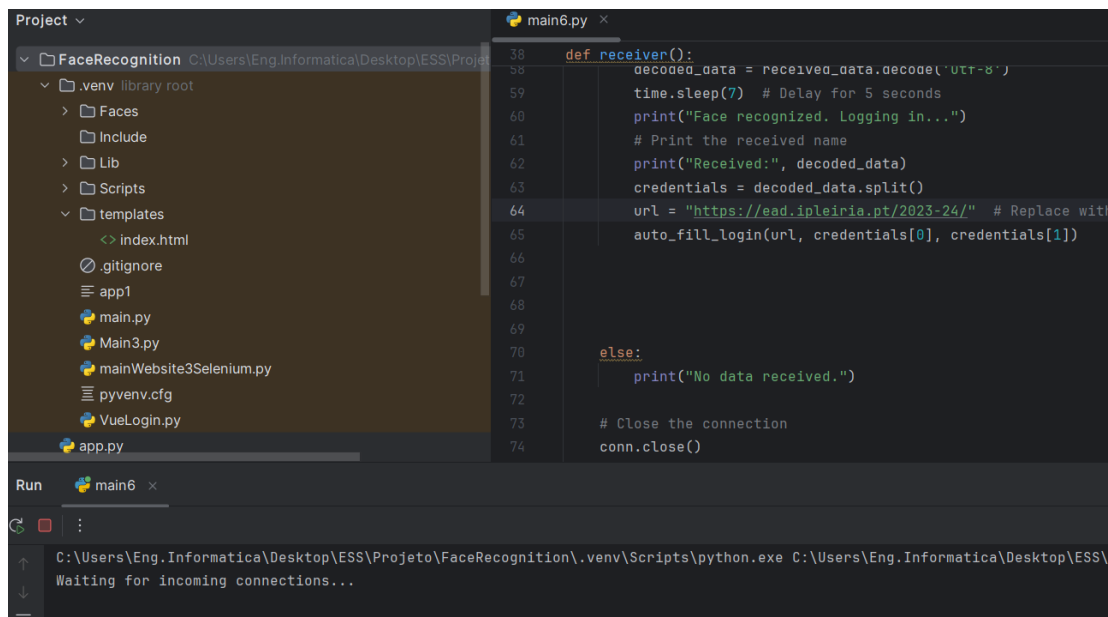
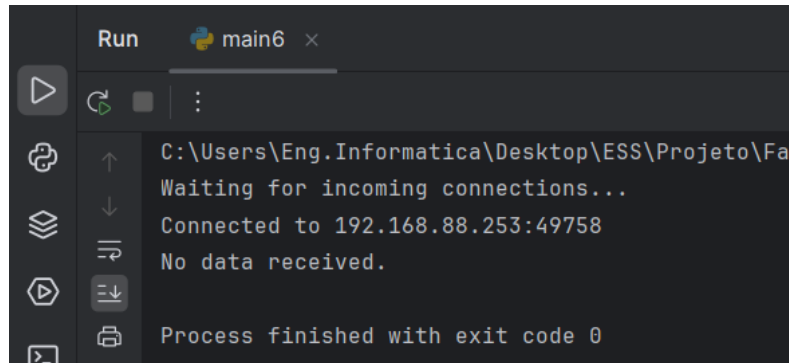


Figure 10 - Vista do Script em execução no computador Receiver

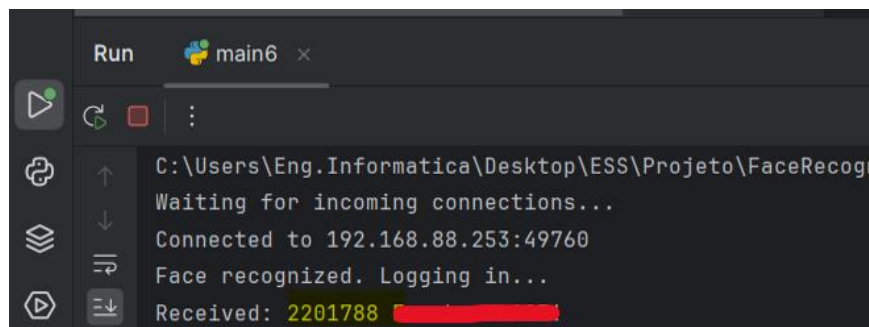
- Se não receber dados válidos apresenta a informação da imagem e termina.



```
Run main6 x
C:\Users\Eng.Informatica\Desktop\ESS\Projeto\Fa
Waiting for incoming connections...
Connected to 192.168.88.253:49758
No data received.
Process finished with exit code 0
```

Figure 11 - Vista do Receiver se não receber dados válidos

- Se o Receiver receber os dados do login com sucesso do utilizador, apresenta a imagem abaixo onde podem ser verificados os dados de login do utilizador.



```
Run main6 x
C:\Users\Eng.Informatica\Desktop\ESS\Projeto\FaceRecogn
Waiting for incoming connections...
Connected to 192.168.88.253:49760
Face recognized. Logging in...
Received: 2201788 [REDACTED]
```

Figure 12- Vista da Receção dos dados de login no Receiver

NOTA: os dados do utilizador são apresentados em plain text, o que representa uma falha de segurança. Num cenário real esta informação deve estar oculta na Base de dados e encriptada.

- Quando recebe essa autenticação, abre o website indicado no código, preenche os campos de login e inicia sessão com as credenciais que são enviadas no teste de comunicação.

```
if received_data:
    decoded_data = received_data.decode('utf-8')
    time.sleep(7) # Delay for 5 seconds
    print("Face recognized. Logging in...")
    # Print the received name
    print("Received:", decoded_data)
    credentials = decoded_data.split()
    url = "https://ead.ipleiria.pt/2023-24/" # Replace with the actual file path
    auto_fill_login(url, credentials[0], credentials[1])
```

Figure 13- Vista do código Python onde esta inserido o website

Como se pode verificar na imagem, o website foi adicionado ao código manualmente, este código está no computador Receiver, que executa este script e fica á espera da autenticação do computador Sender

- Após autenticação do utilizador com sucesso, e envio dos dados entre o Sender e o Receiver, o computador Receiver abre página web indicada no código Python e procede ao login.

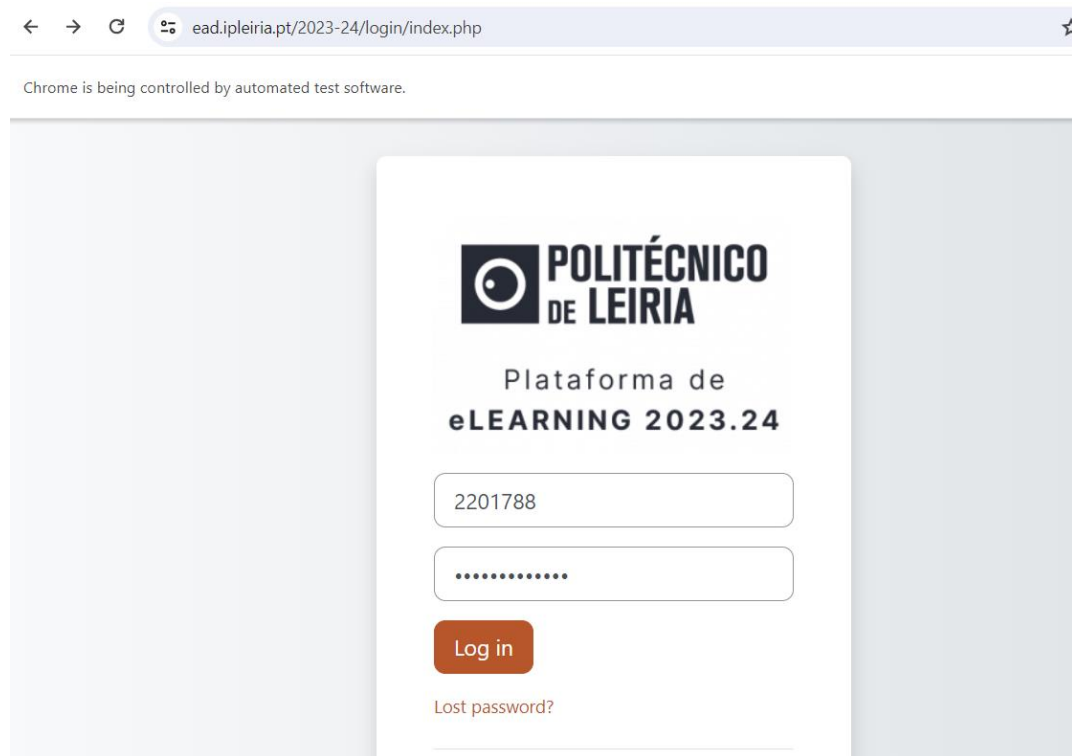


Figure 14 -Vista do login no website

- Não está implementada a Autenticação Multifator após o login, aumentando o risco para a segurança dos dados dos utilizadores.
- O acesso está restrito apenas ao website especificado no código,
- O sistema não é capaz de contornar websites que utilizem sistemas de CAPTCHA.

Manual de Instalação e configuração

É recomendável executar as aplicações de envio e receção em ambiente Linux, dado que o pacote "face_recognition" apresenta melhor desempenho e maior compatibilidade neste tipo de sistema operativo.

Passos para a configuração:

1º Garantir a disponibilidade de dois computadores, podendo ser Windows ou Linux (por exemplo, Ubuntu Desktop), com as respetivas aplicações instaladas.

2º Proceder à instalação dos seguintes pacotes:

- CV2 - Face_Recognition - OS - Socket - Time - Selenium

3º Estabelecer uma rede privada entre os dois computadores, utilizando um router ou switch.

4º No computador destinado a executar a aplicação de envio (Sender), é necessário editar o código para incluir no parâmetro "receiver_ip" o endereço IP do computador que executa a aplicação de receção (Receiver).

5º Criar uma pasta para armazenar as fotos dos utilizadores.

6º Criar uma pasta para armazenar as credenciais de login dos utilizadores em ficheiros TXT, onde as fotos e os ficheiros de texto com as respetivas credenciais devem possuir o mesmo nome, por exemplo: raul.jpg e raul.txt.

As credenciais devem seguir o formato (username password) no ficheiro TXT.

7º O computador que executa a aplicação Receiver deve ser o primeiro a iniciar o ficheiro e permanecer à espera de comunicação do Sender.

8º O computador Sender inicia o código, ativa a Webcam, identifica o utilizador e envia as informações para o computador Receiver.

9º O computador Receiver recebe as credenciais, abre a página Web, preenche os campos de login e inicia sessão no portal.

Conclusões

- Pretende-se com este trabalho, disponibilizar aos utilizadores, uma tecnologia que lhes permita fazer login de forma segura sem necessidade de inserção de username e password, por reconhecimento facial.
- Foi possível testar a tecnologia, com limitações, utilizando dois computadores.
- Foi possível identificar os utilizadores, através de reconhecimento facial por webcam.
- Foi possível detetar utilizadores que não são reconhecidos pela webcam.
- Foi possível autenticar no website de forma automática após reconhecimento facial do utilizador.
- Deve existir boa iluminação no local para que a leitura facial do utilizador seja feita.
- Os níveis de segurança do protótipo não são os adequados para uma aplicação funcional, devem ser melhorados em futuras implementações.

Bibliografia

Face Recognition

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library.

<https://pypi.org/project/face-recognition/>

Keep OpenCV Free

OpenCV is raising funds to keep the library free for everyone, and we need the support of the entire community to do it.

[opencv-python · PyPI](#)

Socket — Low-level networking interface

This module provides access to the BSD *socket* interface. It is available on all modern Unix systems, Windows, MacOS, and probably additional platforms.

[socket — Low-level networking interface — Python 3.12.3 documentation](#)

Python language bindings for Selenium WebDriver.

The *selenium* package is used to automate web browser interaction from Python.

[selenium · PyPI](#)