

Trabalho final - Quantum Finance para Cloud & Cognitive

Cloud and Cognitive Environment O setor de fraudes apontou que existem clientes que se queixaram de não contratar serviços específicos, como o crédito pessoal. Entretanto após o indicador de Detecção de vivacidade (liveness), desenvolvido na disciplina de Computer Vision, ter apresentado um percentual de vivacidade menor que 90% apontou a necessidade de uma nova validação do self da pessoa com o documento.



1. Extração da Face, Nome e CPF



2. Comparação de faces (precisam ter mais do 90% de semelhança)



3. Extração do Nome e Endereço. Nome precisa ser o mesmo da CNH.

Como será a avaliação? O grupo será avaliado se consegue explorar soluções prontas e genéricas de IA disponíveis na nuvem para ajudar a construir soluções de valor para a empresa. Neste processo, o grupo deverá explorar soluções existentes nas principais nuvens, identificar e lograr ativação de serviços, fazer integrações por API's, manipular dados semi-estruturados (Json), integrar com códigos próprios em Python.- Utilização de um ou mais serviços que permitam extração de características de documentos (30% da nota)- Utilização de um ou mais serviços que permitam comparação de faces (20% da nota)- Utilização de um ou mais serviços que permitam extração de informações de comprovante de endereço (20% da nota)- Integração lógica de todos os serviços (30% da nota) Fomatada entrega:- Conjunto de códigos (Python ou Notebook) e documento Word com a explicação do que o grupo realizou

QuantumFinance – Validação Biométrica de Documentos (Google Cloud Streamlit)

Contexto

Este projeto implementa um **pipeline de validação biométrica** para documentos (CNH, RG, comprovantes) utilizando:

- **Google Cloud Vision API** → Extração de texto (OCR) e detecção facial.
- **ImageHash (pHash/wHash)** → Comparação de similaridade entre selfie e foto extraída do documento.
- **Streamlit** → Aplicação web interativa para upload de arquivos e exibição dos resultados.
- **Pandas / JSON** → Consolidação e exportação dos resultados.

O sistema é capaz de:

- Extrair texto de documentos com OCR.
- Detectar e recortar o rosto da CNH.
- Comparar a selfie enviada com o rosto extraído.
- Validar se o nome no comprovante corresponde ao da CNH.

Trabalho final - Quantum Finance para Cloud & Cognitive

- Gerar relatórios em **JSON**, **CSV** e visualização em **Streamlit**.

Estrutura do Projeto

trabalho_final/

```

|— app/
|   └─ streamlit_app.py    # Aplicação web interativa
|— src/
|   └─ utils.py           # Funções de OCR, face extraction e comparação
|— data/
|   └─ 002.JPG            # CNH de teste
|   └─ 003.jpg            # Conta/Comprovante
|   └─ LUIZ.png           # Selfie válida
|   └─ Maria.png          # Selfie inválida
|— outputs/               # Resultados gerados (JSON, CSV, métricas, imagens)
|— cred/
|   └─ dts-10-ds-xxxx.json # Chave de API do Google Vision
|— main.py                # Pipeline em batch (OCR + Face Match + Export)
|— requirements.txt        # Dependências do projeto
|— README.md              # Documentação

```

Instalação

1. Clonar o repositório

```
git clone https://github.com/seu-usuario/validador-biometrico.git
```

```
cd validador-biometrico
```

2. Criar ambiente virtual

```
python -m venv .venv
```

```
source .venv/bin/activate # Linux / Mac
```

```
.venv\Scripts\activate    # Windows
```

3. Instalar dependências

```
pip install -r requirements.txt
```

4. Configurar Google Cloud Vision

Trabalho final - Quantum Finance para Cloud & Cognitive

- Criar um projeto no [Google Cloud Console](#).
- Ativar a **Vision API**.
- Gerar uma chave de serviço JSON e salvar em cred/dts-10-ds-xxxx.json.
- Definir a variável de ambiente:

```
export GOOGLE_APPLICATION_CREDENTIALS="cred/dts-10-ds-xxxx.json"
```

ou no Windows PowerShell:

```
setx GOOGLE_APPLICATION_CREDENTIALS "cred\dts-10-ds-xxxx.json"
```

Como Usar

Rodar o pipeline batch (main.py)

```
python main.py
```

Isso gera **JSON/CSV** em outputs/ com os resultados da comparação facial e OCR.

Rodar aplicação web (Streamlit)

```
streamlit run app/streamlit_app.py
```

Interface interativa para upload de CNH, comprovante e selfie.

Resultados

Exemplo de saída em JSON (outputs/results_LUIZ.json):

```
{
  "documento_nome": "LUIZ ANTONIO DE OLIVEIRA",
  "documento_cpf": "076.763.758-51",
  "comprovante_nome": "LUIZ ANTONIO DE OLIVEIRA",
  "comprovante_endereco": "R JOSE BASILIO GAMA 65, JACAREI/SP",
  "face_match": true,
  "similaridade": 0.92,
  "nome_valido": true
}
```

Métricas

O sistema gera métricas utilizando a similaridade de cosseno, comparando a imagem original da cnh com a foto selfie a ser analisada.

Trabalho final - Quantum Finance para Cloud & Cognitive

Tecnologias

- Python 3.10+
- Google Cloud Vision API
- Pandas, Numpy
- ImageHash (pHash/wHash)
- Streamlit
- Seaborn / Matplotlib

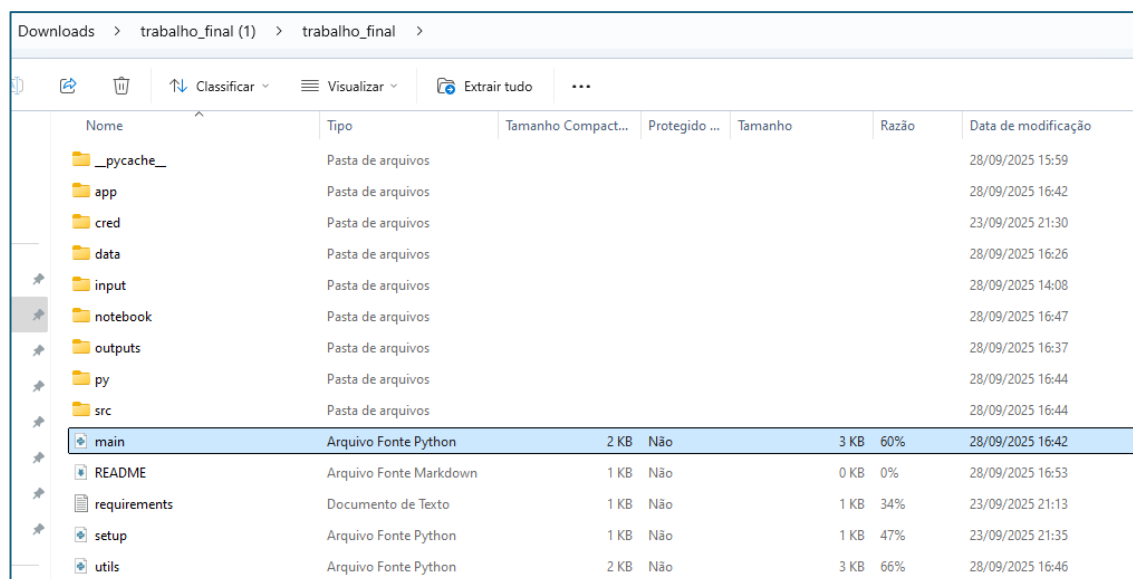
Próximos Passos

- Melhorar comparação facial com modelos de embeddings (ex.: face_recognition, DeepFace).
- Criar banco de dados para armazenar históricos de validações.
- Adicionar testes unitários (pytest).
- Deploy da aplicação Streamlit em **Streamlit Cloud** ou **GCP App Engine**.

Observamos o quão importante foi trabalhar com imagens em boas qualidade, isso foi fator crucial para o resultado da acurácia.

Passo a passo detalhado:

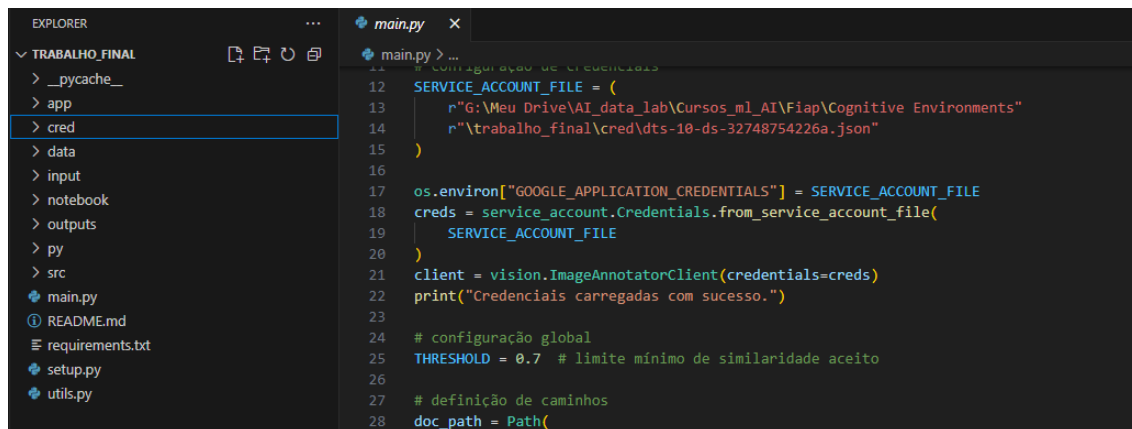
Para executar o projeto deve-se abrir a pasta e abrir o arquivo *main.py*.



Nome	Tipo	Tamanho Compact...	Protegido ...	Tamanho	Razão	Data de modificação
__pycache__	Pasta de arquivos					28/09/2025 15:59
app	Pasta de arquivos					28/09/2025 16:42
cred	Pasta de arquivos					23/09/2025 21:30
data	Pasta de arquivos					28/09/2025 16:26
input	Pasta de arquivos					28/09/2025 14:08
notebook	Pasta de arquivos					28/09/2025 16:47
outputs	Pasta de arquivos					28/09/2025 16:37
py	Pasta de arquivos					28/09/2025 16:44
src	Pasta de arquivos					28/09/2025 16:44
main	Arquivo Fonte Python	2 KB	Não	3 KB	60%	28/09/2025 16:42
README	Arquivo Fonte Markdown	1 KB	Não	0 KB	0%	28/09/2025 16:53
requirements	Documento de Texto	1 KB	Não	1 KB	34%	23/09/2025 21:13
setup	Arquivo Fonte Python	1 KB	Não	1 KB	47%	23/09/2025 21:35
utils	Arquivo Fonte Python	2 KB	Não	3 KB	66%	28/09/2025 16:46

Trabalho final - Quantum Finance para Cloud & Cognitive

Após aberto arquivo importar a pasta, onde o arquivo app será para execução do streamlit, o arquivo de data será onde estão as imagens utilizadas, input foram as imagens que foram utilizadas como teste.

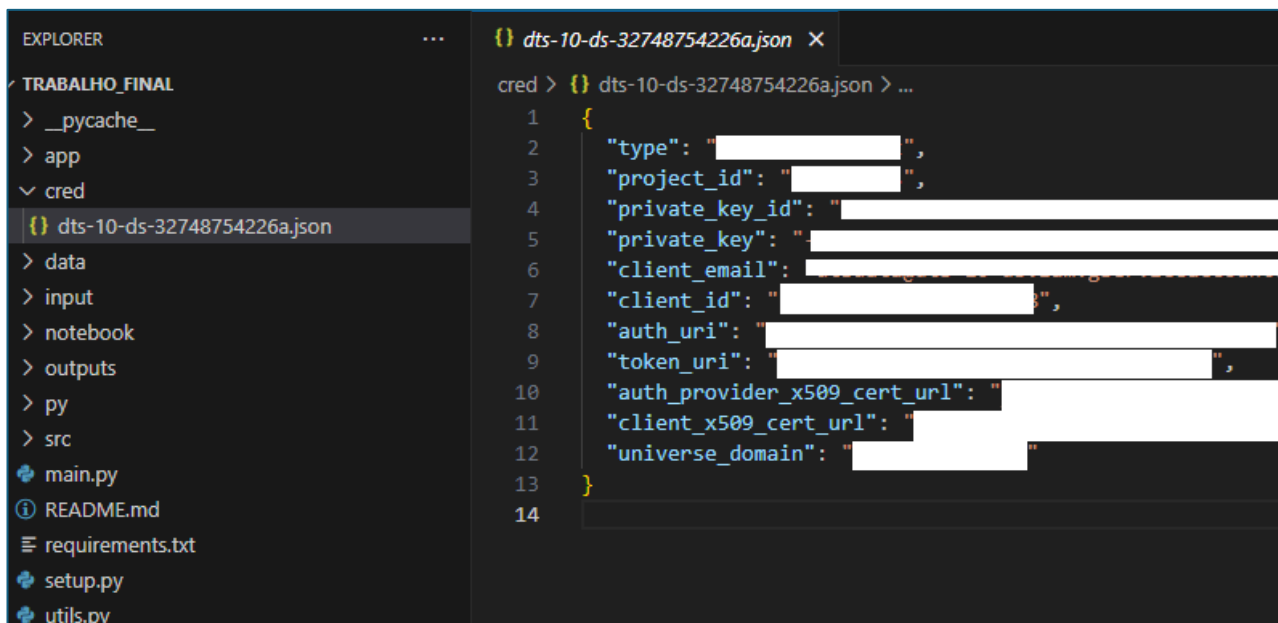


```

11 # configuração de credenciais
12 SERVICE_ACCOUNT_FILE = (
13     r"G:\Meu Drive\AI_data_lab\Cursos_ml_AI\Fiap\Cognitive Environments"
14     r"\trabalho_final\cred\dts-10-ds-32748754226a.json"
15 )
16
17 os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = SERVICE_ACCOUNT_FILE
18 creds = service_account.Credentials.from_service_account_file(
19     SERVICE_ACCOUNT_FILE
20 )
21 client = vision.ImageAnnotatorClient(credentials=creds)
22 print("Credenciais carregadas com sucesso.")
23
24 # configuração global
25 THRESHOLD = 0.7 # limite mínimo de similaridade aceito
26
27 # definição de caminhos
28 doc_path = Path(

```

Para conectar ao GPC, deverá ser gerada sua chave pessoal e inserida no arquivo cred em formato json.



```

cred > {} dts-10-ds-32748754226a.json > ...
1  {
2      "type": "service_account",
3      "project_id": "dts-10-ds-32748754226a",
4      "private_key_id": "dts-10-ds-32748754226a-private-key-id",
5      "private_key": "-----BEGIN PRIVATE KEY-----\n",
6      "client_email": "dts-10-ds-32748754226a@cloudplatform.gcp",
7      "client_id": "dts-10-ds-32748754226a-client-id",
8      "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9      "token_uri": "https://accounts.google.com/o/oauth2/token",
10     "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11     "client_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
12     "universe_domain": "googleapis.com"
13 }
14

```

Após isso executar a aplicação do streamlit que se encontra no arquivo.

Trabalho final - Quantum Finance para Cloud & Cognitive

```

1 import os
2 from pathlib import Path
3 import streamlit as st
4 import pandas as pd
5 from google.cloud import vision
6 from google.oauth2 import service_account
7 from PIL import Image
8 import imagehash
9
10
11 # OCR com Google Vision
12 def extract_text(client, image_path: str) -> str:
13     """Extraí texto de uma imagem usando a API do Google Vision."""
14     with open(image_path, "rb") as f:
15         content = f.read()
16         image = vision.Image(content=content)
17         response = client.text_detection(image=image)
18         texts = response.text_annotations
19         return texts[0].description if texts else ""
20
21
22 # Extração de rosto usando Google Vision
23 def extract_face_and_save(client, image_path: str, output_file: str):
24     """Extraí o rosto de uma imagem e salva em arquivo."""
25     with open(image_path, "rb") as f:
26         content = f.read()
27         image = vision.Image(content=content)
28         response = client.face_detection(image=image)
29         faces = response.face_annotations
30         if not faces:
31             return None
32         face = faces[0]
33         x_min = min(v.x for v in face.bounding_poly.vertices)
34         y_min = min(v.y for v in face.bounding_poly.vertices)
35         x_max = max(v.x for v in face.bounding_poly.vertices)
36         y_max = max(v.y for v in face.bounding_poly.vertices)
37
38         with Image.open(image_path) as pil_img:
39             face_crop = pil_img.crop((x_min, y_min, x_max, y_max))
40             face_crop.save(output_file)
41             return output_file
42
43
44 # Comparação facial via ImageHash (wavelet hash)
45 def compare_faces(img1_path: str, img2_path: str, threshold: float = 0.7):
46     """Compara duas imagens faciais usando wavelet hash."""
47     try:
48         img1 = Image.open(img1_path).convert("L").resize((256, 256))
49         img2 = Image.open(img2_path).convert("L").resize((256, 256))
50         hash1 = imagehash.wavelet(img1)
51         hash2 = imagehash.wavelet(img2)
52         distance = hash1 - hash2
53         return distance < threshold
54     except:
55         return False

```

Abrir o terminal e executar o comando abaixo:

```

PS C:\Users\tarsisft\Desktop\trabalho_final> streamlit run app/streamlit_app.py

```

Após isso será executada a aplicação onde poderão ser incluídos os arquivos.

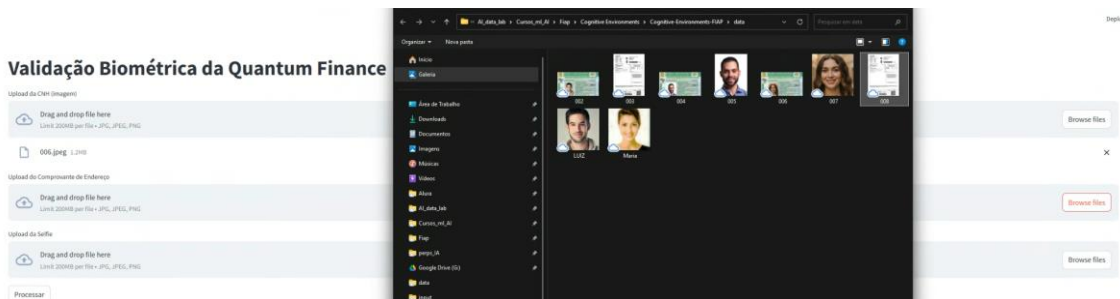


Trabalho final - Quantum Finance para Cloud & Cognitive

Para simular a aplicação com **resultado positivo**, enviar na tela, clicar na opção de busca pela CNH da Neusa e inserir a imagem número 006.



Depois inserir o comprovante que é o número 008



Na sequência inserir a Selfie que é imagem 007:



Clicar no botão de processar para que a aplicação possa funcionar:

Trabalho final - Quantum Finance para Cloud & Cognitive



Processar

Executando OCR...

Extraindo rosto da CNH...

Comparando selfie com CNH...

Resultado

```
{
  "face_match": true
  "similaridade": 0.75
  "documento_extraido": "FODO O TERRITORIO NACIONAL
000000
REPUBLICA FEDERATIVA DO BRASIL
MINISTERIO DA INFRAESTRUTURA
SECRETARIA NACIONAL DE TRANSITO
BR
CARTEIRA NACIONAL DE HABILITAÇÃO/DRIVER LICENSE/PERMISO DE CONDUCCIÓN
2e1 NOME E SOBRENOME
LUIZ ANTONIO DE OLIVEIRA
3 DATA, LOCALE UP DE NASCIMENTO
19/09/1981 SAO PAULO/SP
1 HABILITAÇÃO
24/05/2022
4a DATA EMISSÃO
24/05/2022
4b VALIDADE
ACC
23/05/2023
P
4c DOC IDENTIDADE/ORG EMISSOR/UF
-----"
```

Dessa forma, ele irá ler as fotos incluídas e todo o conteúdo dentro do comprovante de residência, e trará o resumo das informações em formato json:



```
disponível em www.vivo.com.br/meuvivo e pode ser
solicitado impresso, de forma permanente ou não.
Caso ainda tenha dúvidas, ligue para nossa Central
de Relacionamento no 183 15 ou acesse
www.vivo.com.br/faleconosco. Pessoas com
necessidades especiais de fala e audição: 142.
Ao realizar o pagamento, confira se o seu nome, endereço e
números de telefone aparecem no boleto. Você também pode
acessar sua fatura no App da Vivo.
www.anatel.gov.br. Recurso de
Importante: mantenha o pagamento em dia e evite a suspensão parcial/total dos serviços e a inclusão nos órgãos de proteção do crédito. Para pagamentos após o vencimento serão cobrados
encargos de 2% juros de 1% ao mês em conta futura. O ressarcimento por inoperância é realizado em conformidade com as Resoluções: Para STFC artigo 32º da Resolução Anatel nº 426/2005; para SCM
artigo 48º da Resolução Anatel nº 614/2013 para TV artigo 8 da Resolução 488/2007. Central de Atendimento Anatel: 1331 (Geral), 1332 (Deficientes Auditivos)
atendimento VIVO, ligue com o protocolo em mãos para 10325142 para pessoas com necessidades especiais de fala/audição.
(229) PA282 - Plano Ilimitado Local/204-Longa Distância Brasil Tudo
Destaque April
KODIA
Nome do Cliente
LUIZ ANTONIO DE OLIVEIRA
Código do cliente
8999 7099 6273
Código para Cadastro de
Débito Automático
899970996273-9
Número da Fatura
1506437752-0
040400000000 2 60330062089 4 99709962731 50643775299 3
Autenticação Mecânica
Data de Vencimento
27/06/2022
Valor a Pagar (R$)
60,33
Pagar
via Pix"
"nome_valido": true
}
```

face_match	similaridade	documento_extraido	comprovante_extraido	nome_valido
0	0.75	FODO O TERRITORIO NACIONAL 000000 REPUBLICA FEDERATIVA DO BRASIL MINISTERIO DA INFRAESTRUTURA SECRETARIA NACIONAL DE TRANSITO	50643775299 3	0

Trabalho final - Quantum Finance para Cloud & Cognitive

O sistema irá ler a informação da CNH coletando nome e CPF e compara o nome do comprovante de residência. Nesse caso ele exibe se o nome da Cnh e comprovante são iguais ou não.

Também ele fará a comparação via similaridade de cossenos entre a foto extraída da CNH e a selfie fornecida.

Imagens enviadas



CNH



Comprovante



Selfie

Rosto Detectado na CNH

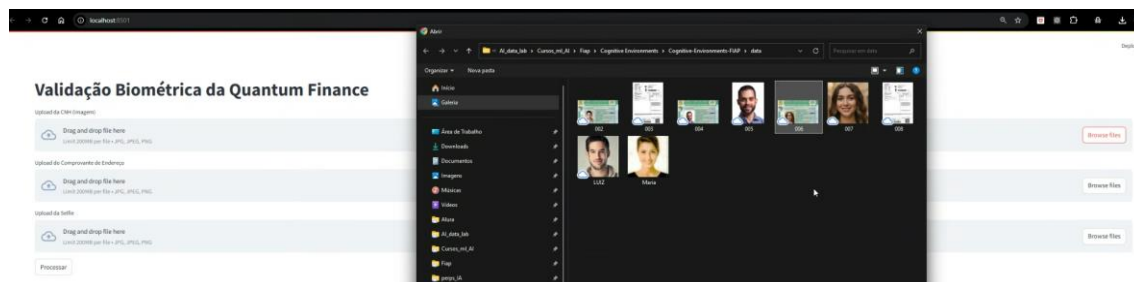


Rosto extraído

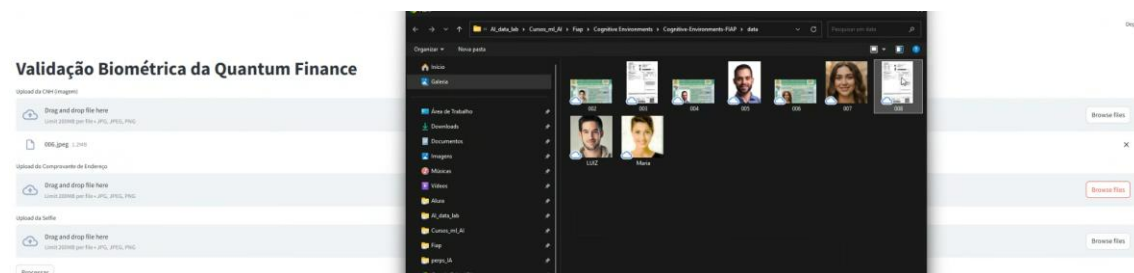
Face compatível! Similaridade: 0.730 (mínimo aceito = 0.7)

Nome da CNH e comprovante são iguais

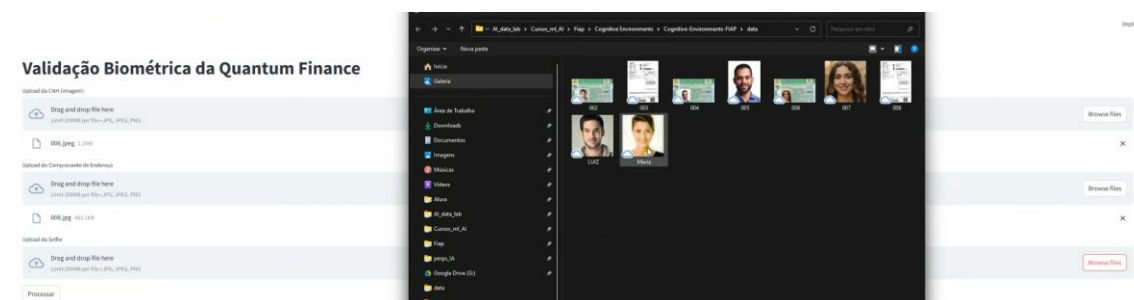
Para o teste que derá errado. Selecionar a CNH de uma pessoa e inserir a selfie/comprovante de outra pessoa.



Subir o comprovante 008



E subir a foto da Maria.



Trabalho final - Quantum Finance para Cloud & Cognitive

Ao processar, vemos o resultado “false”:

Processar

Executando OCR...

Extraindo rosto da CNH...

Comparando selfie com CNH...

Resultado

```

{
  "face_match" : false
  "similaridade" : 0.5
  "documento_extraido" : "FODO O TERRITORIO NACIONAL
                          000000
                          REPUBLICA FEDERATIVA DO BRASIL
                          MINISTERIO DA INFRAESTRUTURA
                          SECRETARIA NACIONAL DE TRANSITO
                          BR
                          CARTEIRA NACIONAL DE HABILITAÇÃO/DRIVER LICENSE/PERMISO DE CONDUCCIÓN
                          2e1 NOME E SOBRENOME
                          LUIZ ANTONIO DE OLIVEIRA
                          3 DATA, LOCALE UP DE NASCIMENTO
                          19/09/1981 SAO PAULO/SP
                          1 HABILITAÇÃO
                          24/05/2022
                          4a DATA EMISSÃO
                          24/05/2022
                          4b VALIDADE
                          ACC
                          23/05/2023
                          P
                          4c DOC IDENTIDADE/ORG EMISSOR/UF
                          513584349 SSPSP
                          4d CPF
                          076.763.758-51
                          NACIONALIDADE
                          BRASILEIRO

```

Podemos verificar que o resultado foi incompatível pois a foto da CNH e a Selfie escolhida são de pessoas diferentes.

Trabalho final - Quantum Finance para Cloud & Cognitive

Imagens enviadas



CNH



Comprovante



Selfie

Rosto Detectado na CNH



Rosto extraído

Face não compatível! Similaridade: 0.500 (mínimo aceito = 0.7)

Nome da CNH e comprovante são iguais

Conclusão

Observamos em vários testes grande importância na qualidade das fotos para que a aplicação consiga reconhecer as diferenças entre as pessoas. Com relação ao o OCR também foi possível verificar as extrações das features e a comparação entre os nomes das CNHs.

Para maiores detalhes estarão disponíveis no github do projeto:

<https://github.com/RafaelGallo/Cognitive-Environments-FIAP/tree/main>.