**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.**

Tela de celular com publicação numa rede social

O conteúdo gerado por IA pode estar incorreto.

**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) Fomatoda 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.
* 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**

* Criar um projeto no [Google Cloud Console](https://console.cloud.google.com/).
* 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.

**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.*

Tabela

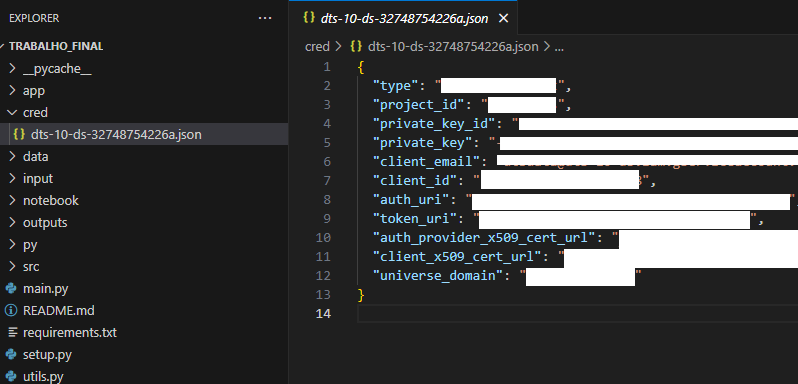
O conteúdo gerado por IA pode estar incorreto.

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.

Texto

O conteúdo gerado por IA pode estar incorreto.

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



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

Texto

O conteúdo gerado por IA pode estar incorreto.

Abrir o terminal e executar o comando abaixo:

Uma imagem contendo Diagrama

O conteúdo gerado por IA pode estar incorreto.

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

Padrão do plano de fundo

O conteúdo gerado por IA pode estar incorreto.

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.

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Depois inserir o comprovante que é o número 008

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Na sequência inserir a Selfie que é imagem 007:

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

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

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

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

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

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.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

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

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Subir o comprovante 008

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

E subir a foto da Maria.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Ao processar, vemos o resultado “false”:

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

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

Texto, Carta

O conteúdo gerado por IA pode estar incorreto.

***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>.