

UNIVERSIDADE FEDERAL DE GOIÁS

CURSO ENGENHARIA DE SOFTWARE

DOMÍNIOS DE SOFTWARE

Sistema de Identificação Visual de Candidatos

ADRIEL LENNER VINHAL MORI

IGOR MOREIRA PÁDUA

MARCOS VINÍCIUS DE MORAES

PAULO ROBERTO VIEIRA

INF

INSTITUTO DE
INFORMÁTICA

2023



Sumário

1. Problema e objetivo
2. Histórias de Usuário e Requisitos
3. Estilo de arquitetura escolhido
4. Decisões arquiteturais
5. Tecnologias escolhidas

Problema e Objetivo

O Desafio é comparar duas imagens, inicialmente estáticas(fotos). Uma imagem estará armazenada em banco de dados, e terá alguns metadados(Nome, CPF, RG, etc.) vinculados à mesma. Ao inserirmos uma nova imagem o sistema deverá efetuar a comparação da "nova" imagem ou imagem de entrada com as existentes no banco de dados, a acurácia deve ser no mínimo de 80%, havendo reconhecimento, o sistema deverá apresentar a imagem reconhecida com as informações da pessoa, se não houver reconhecimento retornar mensagem explicando que não houve reconhecimento.

O objetivo é criar uma IA utilizando código em python que analisa e compara duas imagens(fotos) para dizer se são as mesmas pessoas nas imagens ou não.

Histórias de Usuário e Requisitos

HU01 Confrontar duas imagens

Responsável: Igor

Prioridade: alta

Como operador do sistema,

Quero que o sistema confronte duas imagens

Para que possa ser validado se existe ou não no banco de dados.

Cenário 1: Identificar uma imagem

Dado que existem imagens cadastradas no sistema,

Quando inserimos uma nova imagem,

Então o sistema identifica se a nova imagem corresponde com alguma existente no banco de dados do sistema e retorna mensagem validando ou não.

HU02 Apresentar Dados

Responsável: Paulo

Prioridade: alta

Como operador do sistema,

Quero que o sistema apresente os dados de uma pessoa

Para que possa ser validado as informações

Cenário 2: Validar dados de uma pessoa cadastrada

Dado que uma imagem foi inserida.

Quando houver o reconhecimento com alguma existente no sistema

Então o sistema deve apresentar os dados da pessoa que teve imagem reconhecida e validada pelo sistema.

Histórias de Usuário e Requisitos

HU03 Status do Candidato

Responsável: Adriel Lenner Vinhal Mori

Prioridade: alta

Como operador do sistema,

Quero adicionar um status referente a aplicação e realização da prova ao candidato que está realizando-a

Para sinalizar candidatos que finalizaram a prova, faltantes e que tiveram a realização da prova interrompida por conta de força maior.

Cenário 3: Adicionar status ao candidato.

Dado que um candidato tenha sido previamente identificado,

Quando a identificação do candidato for validada

Então poderá ser adicionado um status referente a participação do candidato na realização da prova - iniciado, indeferido, interrompido, concluído.

HU04 Conferência de Sala

Responsável: Marcos Vinícius de Moraes

Prioridade: alta

Como operador do sistema,

Quero poder adicionar observações sobre o candidato identificado

Para ter um maior controle de situações atípicas que venham a acontecer.

Cenário 4: adicionar observações sobre o candidato.

Dado que um candidato tenha sido previamente identificado.

Quando a identificação do candidato for validada

Então podem ser feitas anotações acerca do candidato e de qualquer irregularidade ou imprevisibilidade ocorrida.

Histórias de Usuário e Requisitos

- Requisitos Funcionais

Prioridade	Identificador	Nome e Descrição do Requisito	Interfaces e Dependências
[1 / 2 / 3]	[TIPSEQ: de acordo com Plano de Gerenciamento de Requisitos.]	[Nome do requisito]: [Detalhes levantados sobre o requisito. Se possível definir o representante do cliente que forneceu a informação.]	Interfaces: [Identificador da interface] Dependências: [Identificador do requisito]
1	RF1	O sistema deve conseguir identificar a mesma pessoa por fotos diferentes	
1	RF2	Após a identificação o sistema deve mostrar os dados do candidato	
1	RF3	Uma vez que a pessoa tenha sido identificada, o sistema possibilita informar o status do candidato - presente, indeferido, concluído.	
1	RF4	Uma vez que a pessoa tenha sido identificada, o sistema permite que seja adicionado uma observação acerca do status da pessoa.	

Histórias de Usuário e Requisitos

- Requisitos Não Funcionais

Prioridade	Identificador	Nome e Descrição do Requisito	Interfaces e Dependências
[1 / 2 / 3]	[TIPSEQ: de acordo com Plano de Gerenciamento de Requisitos.]	[Nome do requisito]: [Detalhes levantados sobre o requisito. Se possível definir o representante do cliente que forneceu a informação.]	Interfaces: [Identificador da interface] Dependências: [Identificador do requisito]
1	RNF1	O sistema deverá ter alta disponibilidade, e estar livre de acessos não autorizados.	
	RNF2	Requisitos decorrentes de políticas e procedimentos corporativos em que o sistema está imerso.	
1	RNF3	O sistema deverá se comunicar com as dependências do SQL Server.	
1	RNF4	Os usuários que administram a plataforma para identificação facial deverão operar o sistema após um determinado tempo de treinamento.	
1	RNF5	A arquitetura do sistema deve estar suscetível a demandas de alteração com tratamento de erros, de modo fácil e que não afetem o funcionamento do sistema.	

Estilo de Arquitetura

- Sistema de Identificação Visual de Candidatos
 - Existem várias abordagens diferentes para a arquitetura de software para IA, exemplos:
 - arquitetura baseada em serviços
 - arquitetura baseada em componentes
 - arquitetura baseada em distribuídas
 - A escolhida foi uma arquitetura baseada em componentes
 - Rede Neural Convolucional (CNN)
 - Processamento e análise inteligente de imagens digitais

Decisões Arquiteturais

Uma arquitetura de software é a estrutura lógica de um sistema de software, que define como os componentes do sistema se comunicam e se relacionam entre si. Uma Arquitetura de Software para Inteligência Artificial pode incluir vários componentes, tais como:

Camada de entrada: Responsável por coletar e processar os dados de entrada para o sistema, como **imagens**, áudio, ou texto.

Camada de modelo: Responsável por armazenar e aplicar o modelo de aprendizado de máquina treinado. Ele pode incluir **redes neurais**, algoritmos de aprendizado supervisionado, entre outros.

Camada de saída: Responsável por produzir a saída do sistema, como previsões, reconhecimento de fala, ou **classificação de imagens**.

Camada de gerenciamento: Responsável por gerenciar as operações do sistema, como monitoramento de desempenho, gerenciamento de recursos, e manutenção.

Camada de usuário: Responsável por prover a interface do usuário com o sistema, como uma interface web ou aplicativo móvel.

Camada de armazenamento: Responsável por armazenar os dados de entrada, saída, e modelos do sistema.

Vale ressaltar que essas camadas são interligadas e trocam informações entre si.

Decisões Arquiteturais

As Decisões Arquiteturais em Inteligência Artificial podem incluir:

Tipo de modelo: Escolha entre modelos de **aprendizado** supervisionado, não supervisionado, semi-supervisionado, entre outros.

Tipo de rede neural: Escolha entre redes neurais feedforward, recorrentes, **convolucionais**, entre outros.

Algoritmo de otimização: Escolha entre algoritmos de otimização como Gradient Descent, Adam, entre outros.

Tamanho do conjunto de treinamento: Escolha do **tamanho** do conjunto de treinamento e do número de épocas de treinamento.

Tipo de dados de entrada: Escolha do tipo de dados de entrada, como **imagens**, áudio, texto, entre outros.

Tipo de saída: Escolha do tipo de saída, como previsões, **classificação**, reconhecimento de fala, entre outros.

Arquitetura de armazenamento: Armazenamento como bancos de dados **relacionais** ou não relacionais.

Arquitetura de computação: Como computação em nuvem ou **computação local**.

Mecanismo de explicação: Tais como LIME, SHAP, entre outros.

Essas decisões afetam o Desempenho, a Escalabilidade, e a Facilidade de manutenção do sistema AI.

Decisões Arquiteturais

Os componentes do sistema devem ser projetados de tal forma que cada um tenha uma única responsabilidade e não se sobreponha a outros.

Escalabilidade: o sistema deve ser projetado para escalar facilmente à medida que o volume de tráfego aumenta.

Manutenibilidade: o sistema deve ser fácil de manter e atualizar, com componentes que podem ser substituídos ou atualizados sem afetar o funcionamento geral do sistema.

Testabilidade: o sistema deve ser projetado de tal forma que os componentes possam ser testados de forma isolada para garantir a confiabilidade do sistema.

Reutilização: os componentes do sistema devem ser projetados de forma que possam ser reutilizados em outros projetos.

Existem muitas arquiteturas de software diferentes que se adaptam para diferentes tipos de sistemas e aplicações, incluindo arquitetura monolítica, arquitetura de microserviços, arquitetura de eventos, arquitetura baseada em componentes e muito mais.

Decisões Arquiteturais

O estilo arquitetural escolhido para um sistema de inteligência artificial dependerá dos requisitos do sistema e das necessidades do projeto. Alguns estilos arquiteturais comuns para sistemas de IA incluem:

Arquitetura baseada em agentes: Agentes inteligentes são os principais componentes do sistema. Os agentes são entidades autônomas que agem para atingir seus objetivos.

Arquitetura baseada em serviços: O sistema é dividido em serviços independentes que podem ser combinados para criar soluções de IA mais complexas.

Arquitetura baseada em componentes: O sistema é dividido em componentes reutilizáveis que podem ser combinados para criar soluções de IA personalizadas.

Arquitetura distribuída: O sistema é executado em múltiplos dispositivos ou computadores conectados em rede. Isso permite escalabilidade e distribuição de processamento.

As decisões arquiteturais incluem como os componentes do sistema se relacionam e como eles se comunicam entre si. Isso inclui a seleção de protocolos de comunicação, padrões de projeto, metodologias de desenvolvimento e outros aspectos relacionados à arquitetura do sistema.

Pipeline of Proposed Technologies

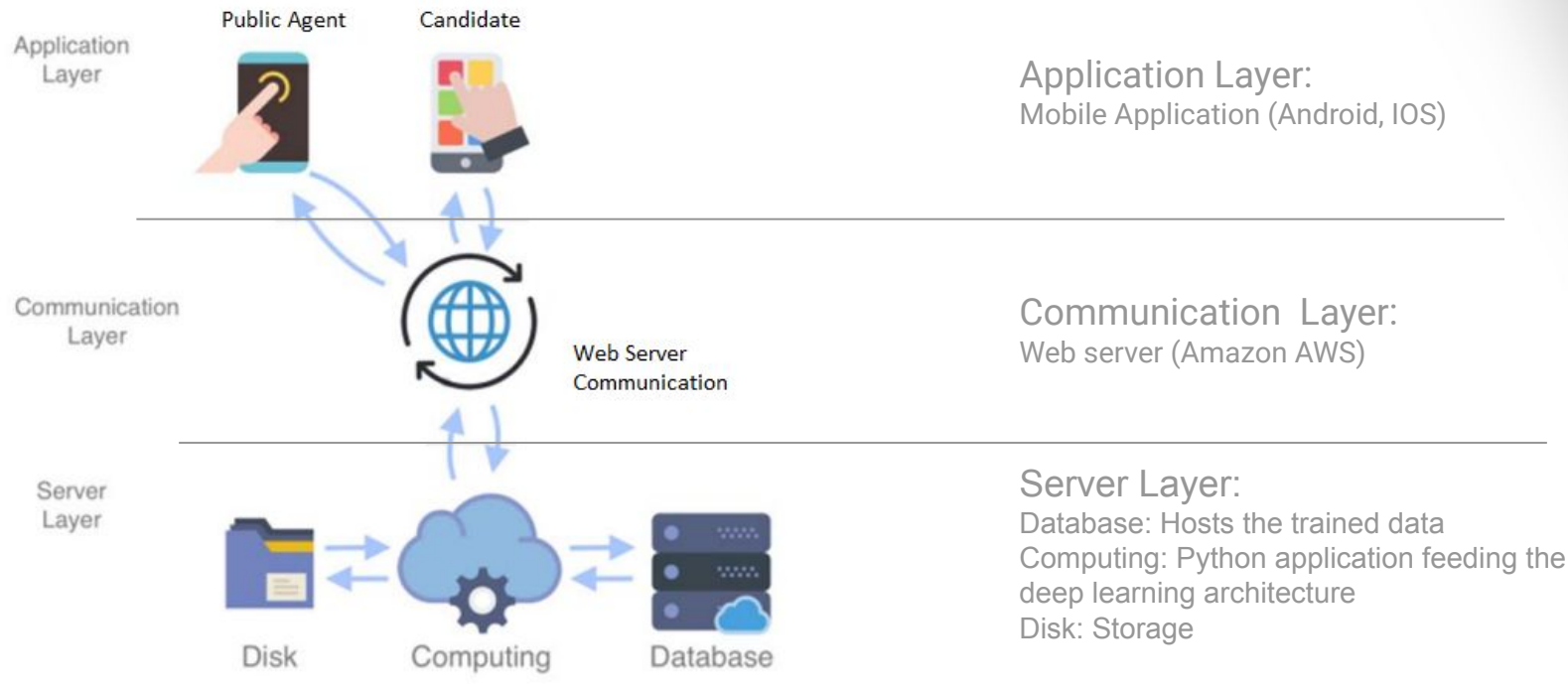
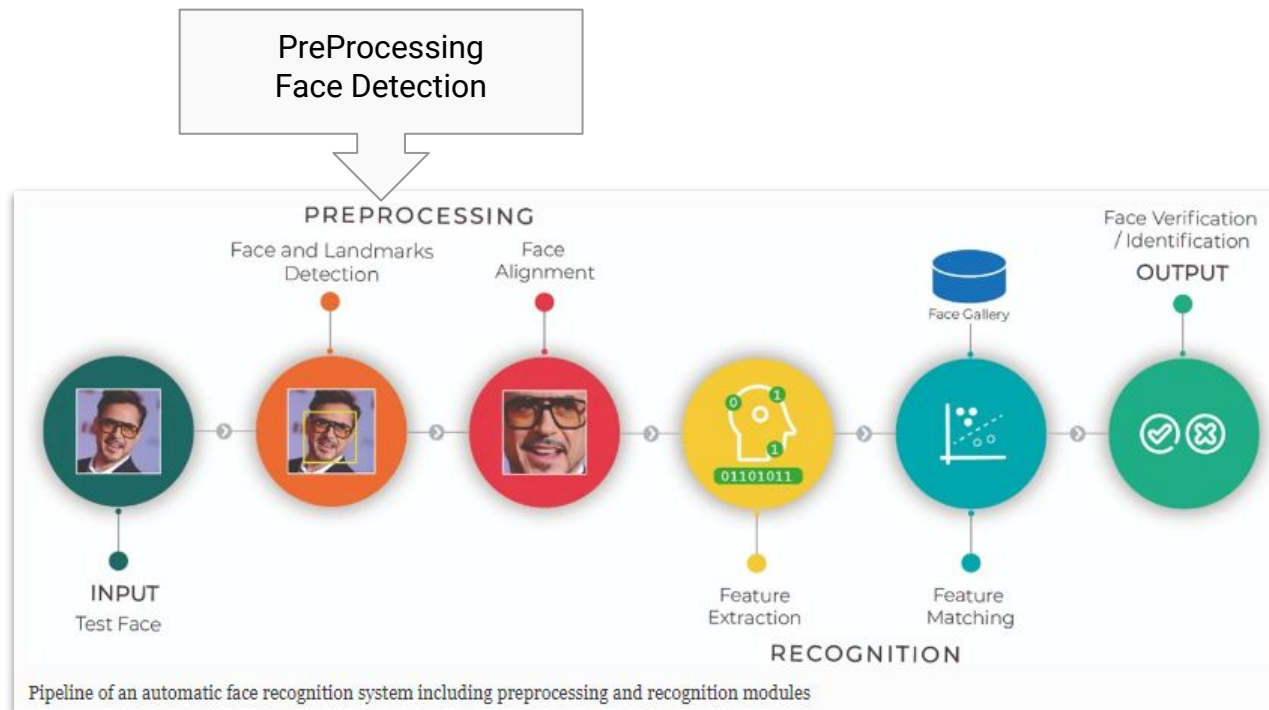
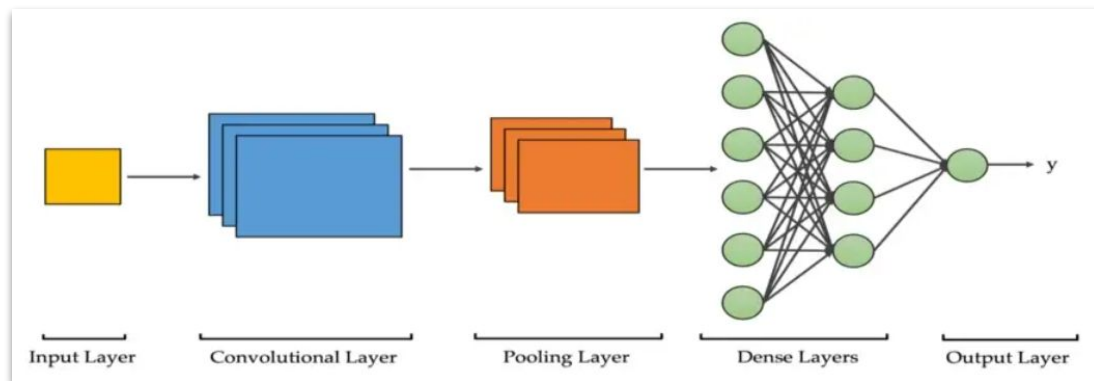
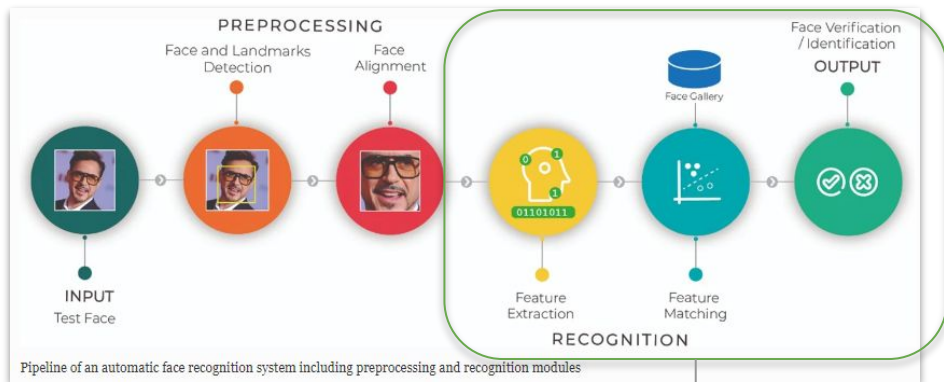


Figure 1. System Architecture

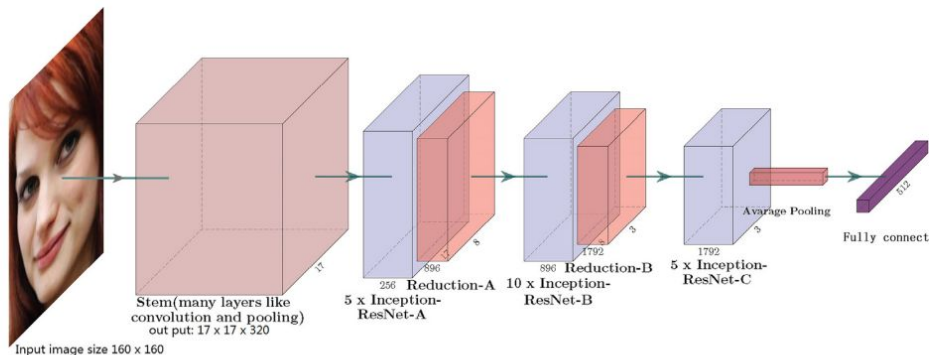
Face Detection



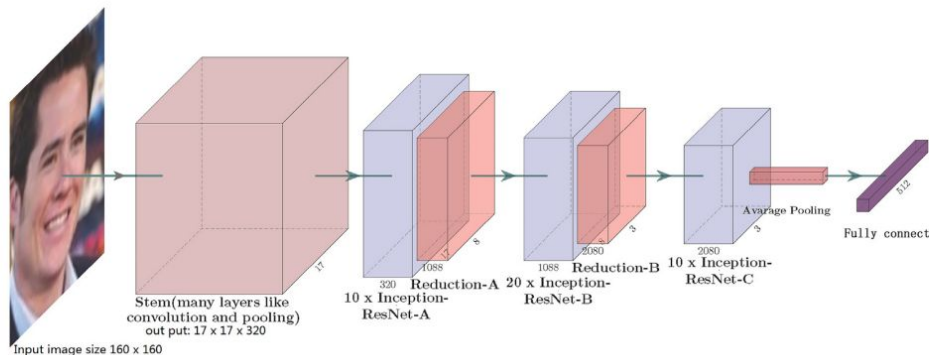
Face Recognition



Proposed Model Recognition Technology



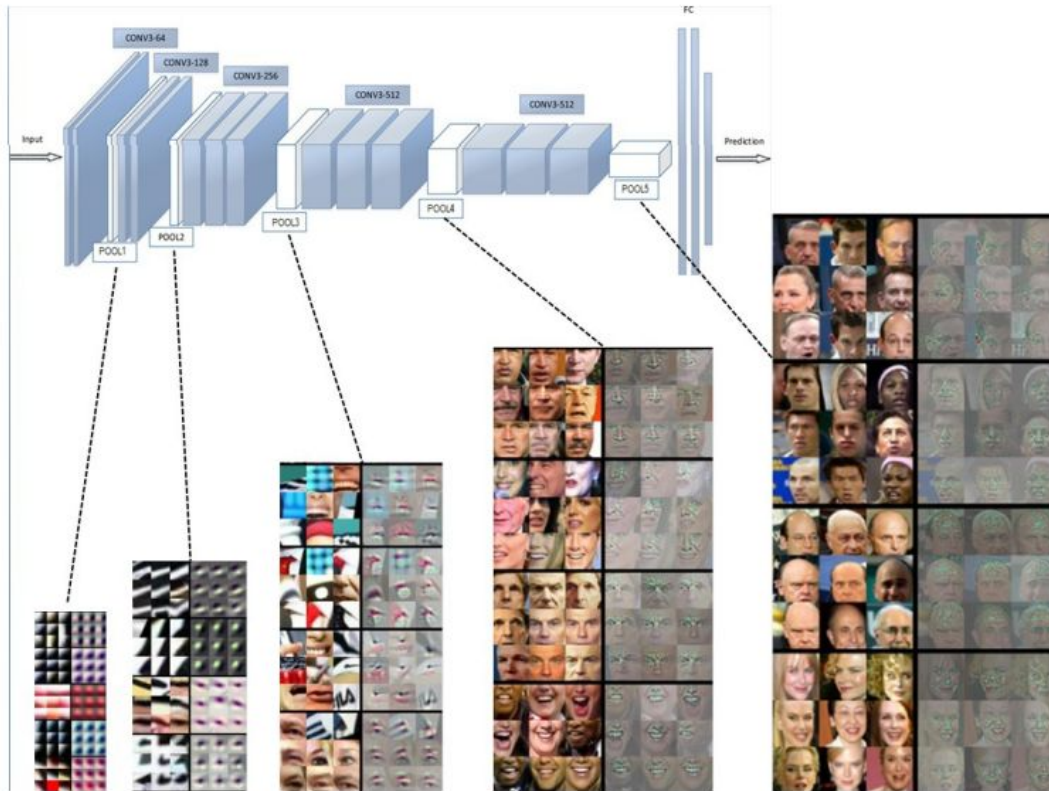
(a) architecture of Inception-ResNet v1



Proposed Model Recognition Technology

Deep face recognition: A survey

Masi, Iacopo, et al. "Deep face recognition: A survey." *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. IEEE, 2018.



Obrigado!

INF
INSTITUTO DE
INFORMÁTICA



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS