**Computação Cognitiva – Projeto Final**

**Miniprojeto de reconhecimento facial**

* **Descrição do projeto**

O reconhecimento facial é umas das subáreas da Inteligência Artificial que tem como objetivo reconhecer faces de pessoas em imagens ou vídeos. Um exemplo são os sistemas de segurança que podem utilizar esses recursos para identificar se uma pessoa está ou não presente em um ambiente. Neste contexto é importante frisar as diferenças entre as técnicas de detecção e reconhecimento facial. Enquanto a primeira somente indica se uma face está presente em uma imagem, a segunda técnica tem o objetivo de dizer de quem é a face detectada.

Com base nisso, este projeto é focado no reconhecimento facial com o intuito de reconhecer faces pela webcam. Utilizaremos a linguagem SQL, Python e a biblioteca OpenCV, que é uma das mais utilizadas para processamento digital de imagens e visão computacional. Faremos testes com o algoritmo LBPH (Local Binary Patterns Histograms) disponibilizado no OpenCV.

* **Objetivo**

Construção de um minissistema de suporte as escolas e faculdades, no qual realiza à chamada em sala de aula através do reconhecimento facial.

* **Motivação**

Este projeto tem por motivação entregar as escolas e faculdades um sistema que realize a chamada sozinho. Espera-se que o tempo perdido na hora da chamada seja refletido em resolução de dúvidas entre outras coisas, tornando por fim as aulas mais produtivas.

* **Fases CRISP-DM**

Este projeto pode ser divido em quatro partes, sendo elas: detecção, coletar os dados (fotos), treinar e por fim reconhecer.

Passando essas partes para as fases do CRISP-DM temos:

1. Entendimento do negócio:

A primeira coisa a ser feita é entender de fato qual o problema a ser resolvido e quais os objetivos, ou seja, a construção de um sistema que realize a chamada em sala de aula sozinho.

1. Entendimento e preparação dos dados:

Essas etapas consistem em organizar todos os dados importantes para a resolução do problema e como eles serão cruzados para resolver o problema. Dentro dessas etapas entra a parte de detecção da face e pôr fim a captura da foto.

1. Modelagem:

É nesta fase que as fotos (Dados) podem ser usadas para alimentar o algoritmo LBPH, ou seja, onde é feito o treinamento

1. Avaliação do desempenho:

Agora entra o que é feito após o trabalho propriamente dito. Trata-se de um momento onde serão acompanhados os resultados em relação aos objetivos, ou seja, onde acompanharemos se o algoritmo está fazendo corretamente o reconhecimento facial.

1. Implantação:

Aqui, é a hora da verdade, onde tudo que foi obtido serão entregues de forma a ser aplicada em faculdades e escolas.

* **Descrição do algoritmo LBPH**

Local Binary Pattern (LBP) é um operador de textura simples, porém eficiente, que rotula os pixels de uma imagem ao limitar a vizinhança de cada pixel e considera o resultado como um número binário. Em outras palavras, o LBPH pega o elemento central de um pixel e compara com o vizinho, formando assim uma matriz binária. A partir da matriz binária de uma face é possível converter todos os valores para decimal e em cima disso é feito um histograma

° Treinamento:

- O número decimal é usado para treinar o sistema, gerando um histograma dos valores (para cada face)

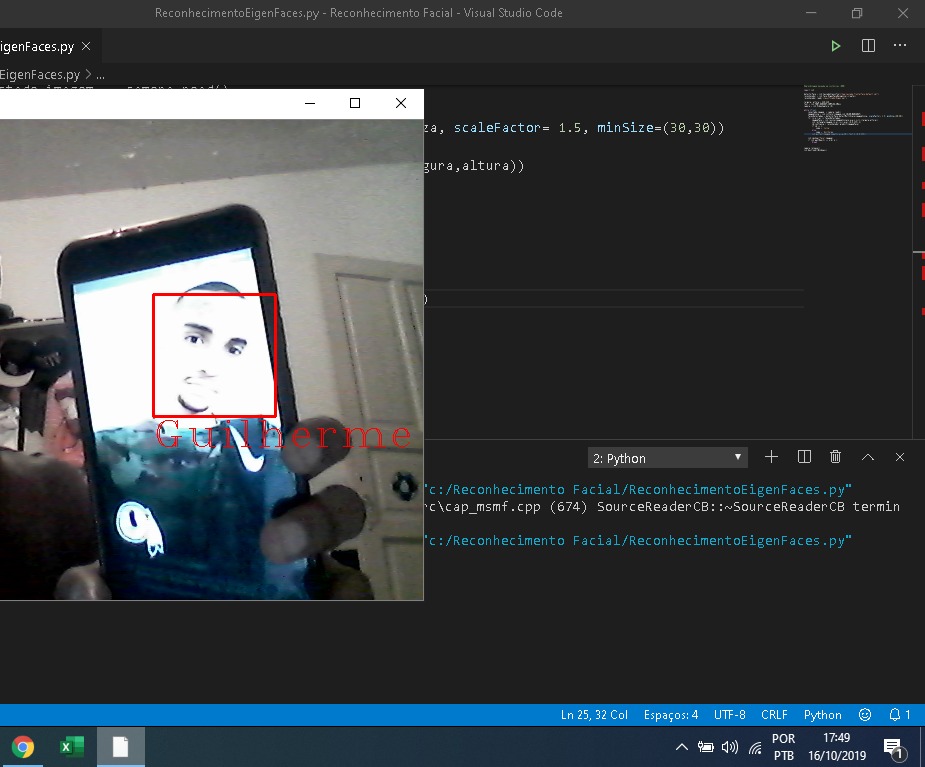
- Algoritmo é caracterizado por encontrar a estrutura local da imagem por meio dos pixels vizinhos

° Classificação:

- Gera o histograma para a nova imagem e compara com os histogramas da base de dados

* **Resultados Obtidos**

Aqui temos apenas o protótipo do sistema de reconhecimento facial, podemos ver que o algoritmo é tão eficaz que consegue fazer o reconhecimento até pela tela de um celular.



* **Conclusão**

O desenvolvimento deste projeto nos permitiu desenvolver uma solução usando linguagens de programação, como Python e SQL, que são constantemente usadas no mercado. Graças a isso, o objetivo foi cumprido e conseguimos chegar ao resultado final, no qual o minissistema de auxílio a chamada poderá realizar o reconhecimento facial dos alunos e marcar "Presente" ou "Falta" no banco de dados.

**Códigos**

Todos os códigos, arquivos e pastas podem ser encontrados através de um repositório no GitHub, basta acessar o seguinte link: <https://github.com/GuiiLuiss/Reconhecimento-Facial>