

UNIVERSIDADE ESTADUAL DO RIO GRANDE SUL  
ENGENHARIA DE COMPUTAÇÃO

ALLEFF DYMYTRY PEREIRA DE DEUS

**RECONHECIMENTO FACIAL PARA  
CLASSIFICAÇÃO E REGISTRO DE  
PRESENÇA EM SALA DE AULA**

Trabalho de Conclusão apresentado como  
requisito parcial para a obtenção do grau de  
Engenheiro em Engenharia de Computação

Profa. Dra. Letícia Vieira Guimarães  
Orientador

Guaíba, Janeiro de 2020

## FOLHA DE APROVAÇÃO

Monografia sob o título "*Reconhecimento Facial para Classificação e Registro de Presença em Sala de Aula*", defendida por Alleff Dymytry Pereira de Deus e aprovada em 3 de Julho de 2020, em Guaíba, estado do Rio Grande do Sul, pela banca examinadora constituída pelos professores:

---

Profa. Dra. Adriane Parraga  
Orientadora

---

Prof. Dr. João Leonardo Fragoso

---

Profa. Dra. Letícia Vieira Guimarães

---

Prof. Dr. Roberto Ribeiro Baldino

*"Você deve entender que há mais de um caminho para o topo da montanha."*

— MIYAMOTO MUSASHI

## **AGRADECIMENTOS**

Aos meus ....

À minha ...

Aos meus...

À Universidade, professores e funcionários que propiciaram um excelente ambiente de aprendizagem.

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b>	<b>6</b>
<b>LISTA DE FIGURAS</b>	<b>7</b>
<b>LISTA DE TABELAS</b>	<b>8</b>
<b>RESUMO</b>	<b>9</b>
<b>ABSTRACT</b>	<b>10</b>
<b>1 INTRODUÇÃO</b>	<b>11</b>
<b>2 FUNDAMENTOS TEÓRICOS</b>	<b>12</b>
2.1 EIGENFACES	12
2.1.1 Reconhecimento Utilizando Eigenfaces	13
2.2 SISTEMAS EMBARCADOS	14
2.2.1 Sons normais	14
<b>3 MATERIAIS E MÉTODOS</b>	<b>16</b>
3.1 MATERIAIS	16
3.2 PYTHON	16
3.3 OPENCV	17
3.4 MONGODB	17
3.5 EIGENFACES	17
3.6 METODOLOGIA	17
<b>4 RESULTADOS E DISCUSSÕES</b>	<b>19</b>
4.1 Obtenção das Eigenfaces	19
4.2 Interface	21
<b>5 CONCLUSÃO</b>	<b>23</b>
<b>6 TRABALHOS FUTUROS</b>	<b>24</b>
<b>REFERÊNCIAS</b>	<b>25</b>

## LISTA DE ABREVIATURAS E SIGLAS

TWP	Transformada <i>Wavelet Packet</i>
PDS	Processamento Digital de Sinais
RNA	Rede Neural Artificial
SLIT	Sistemas Lineares e Invariantes no Tempo
FPA	Filtro Passa-Alta
FPB	Filtro Passa-Baixa
FIR	<i>Finite Impulse Response</i> (Resposta ao Impulso Finita)
IIR	<i>Infinite Impulse Response</i> (Resposta ao Impulso Infinita)
ADC	<i>Analog-to-digital converter</i> (Conversor Analógico-Digital)
TF	Transformada de Fourier
TDF	Transformada Discreta de Fourier
STFT	<i>Short Time Fourier Transform</i> (Transformada de Fourier de Tempo Reduzido)

## LISTA DE FIGURAS

2.1	Esquemático do método de classificação. . . . .	12
2.2	Espectros dos sinais: (a) sinal $X(f)$ , (b) trem de impulsos, (c) sinal $X(f)$ multiplicado pelo trem de impulsos. . . . .	15
3.1	Diagrama das etapas de desenvolvimento. . . . .	18
3.2	Diagrama do sistema. . . . .	18
4.1	Face pertencente ao treino. . . . .	19
4.2	Face média $\Psi$ . . . . .	20
4.3	Faces $\Phi$ . . . . .	20
4.4	Eigenfaces. . . . .	21
4.5	Interface do sistema. . . . .	22

# LISTA DE TABELAS

2.1	Tipos de sons normais. . . . .	14
-----	--------------------------------	----



## **RESUMO**

Este trabalho tem por objetivo o desenvolvimento de um sistema que detecta faces para o auxílio no registro de alunos presentes no ambiente acadêmico. Para a localização facial será utilizado o método de classificação Haar Like Features e para a classificação das faces encontradas será utilizado o método Eigen Faces. Para a facilidade de utilização do usuário final, os métodos irão ser integrados em um sistema que possibilita a inclusão dos dados diretamente no banco de dados, possibilitando a consulta dos mesmo com mais agilidade.

**Palavras-chave:** Eigenfaces, Reconhecimento Facial, Haar Like Features.

## **Facial Recognition for Classification and Register on Presence in Classroom**

### **ABSTRACT**

(COLOCAR EM INGLES DEPOIS)Este trabalho tem por objetivo o desenvolvimento de um sistema que detecta faces para o auxílio no registro de alunos presentes no ambiente acadêmico. Para a localização facial será utilizado o método de classificação Haar Like Features e para a classificação das faces encontradas será utilizado o método Eigen Faces. Para a facilidade de utilização do usuário final, os métodos irão ser integrados em um sistema que possibilita a inclusão dos dados diretamente no banco de dados, possibilitando a consulta dos mesmo com mais agilidade.(COLOCAR EM INGLES DEPOIS)

**Keywords:** key words.

# 1 INTRODUÇÃO

Nos dias atuais pode-se notar que a biometria é algo amplamente utilizado e ajuda no cotidiano de todas as pessoas, já que pode-se fazer as mais diversas funções, sendo elas: pagamento de contas, liberação para áreas restritas, verificação em e-mails, para direção de um veículo, etc. A biometria tem por base a medição de características humanas de forma analógicas e transforma para o mundo digital, sendo uma delas a mais emergente o reconhecimento facial.

Segundo a ANSA, no carnaval de 2019, a polícia do Rio de Janeiro e do Salvador, conseguiram detectar e prender criminosos com a ajuda de câmeras equipadas com reconhecimento facial. Existe no mercado atual um crescimento de 20

O reconhecimento facial pode ser empregado nas mais diferentes áreas, sem ser a da segurança, com isso pode-se empregar o reconhecimento para verificar sentimentos, expressões, executar comandos configurados, check-in em eventos, entre outras aplicações.

Com esses avanços nas técnicas de reconhecimento facial, pode-se utilizar tais avanços para o reconhecimento de alunos em sala de aulas, já que os mesmos querem ter seus rostos reconhecidos para obter a presença em sala de aula, fazendo que o método tradicional de folha de chamada possa ser substituído e todo o sistema de presença seja diretamente integrado em um único sistema.

Desta forma, a construção de um sistema de reconhecimento facial integrado com um controle de presença se faz necessário para sanar o problema de ainda hoje, em um tempo de integração e avanço tecnológico, utilizar chamadas impressas para marcar se o aluno estava ou não em aula ou em um determinado evento.

## 2 FUNDAMENTOS TEÓRICOS

Neste capítulo serão abordados os conhecimentos necessários para a contextualização do presente trabalho, bem como seus métodos de desenvolvimento e funcionalidades específicas. O processo de funcionamento do presente trabalho segue a ordem indicada na figura 2.1

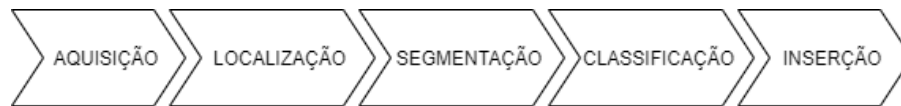


Figura 2.1: Esquemático do método de classificação.

- **AQUISIÇÃO:** O processo de aquisição se refere ao momento em que o usuário tem sua face adquirida por um dispositivo de captura (câmera, filmadora, webcam, etc.);
- **LOCALIZAÇÃO:** O processo de localização tem por funcionamento encontrar uma face na imagem adquirida no processo anterior;
- **SEGMENTAÇÃO:** O processo de segmentação por sua vez retira para o sistema somente a face localizada no processo anterior, facilitando a próxima etapa do sistema;
- **CLASSIFICAÇÃO:** O processo de classificação consiste em utilizar a face para o reconhecimento de qual usuário está no sistema;
- **INSERÇÃO:** O processo de inserção finaliza o sistema, inserindo qual usuário que foi reconhecido pelos processos anteriores.

### 2.1 EIGENFACES

No contexto de análise de imagens, existe um problema em qualquer tipo de classificação, sendo ele o tamanho das dimensões da imagem, onde um problema simples pode se tornar complexo, já que uma imagem em escala de cinza de 300 pixels por 300 pixels irá possuir um total de 90000 valores diferentes ( $m = i \times j$ ). Sendo assim é necessário retirar das imagens somente o que interessa, para isso é utilizado o método de análise de componentes principais (PCA), onde serão extraídas das imagens as partes que possuem as maiores relevâncias dentro do espaço de distribuição de dados.

O PCA é feito pela decomposição em autovetores de uma matriz de covariância, sendo o autovetor o maior autovalor associado correspondente do da componente principal de cada conjunto.

### 2.1.1 Reconhecimento Utilizando Eigenfaces

Pode-se utilizar eigenfaces para reconhecimento facial, alguns passos devem ser feitos antes do reconhecimento concreto, levando em consideração que as imagens utilizadas estão centralizadas e possuem os mesmos tamanhos. Sendo assim pode-se dividir o processo em duas etapas, a primeira de treinamento e o segundo de reconhecimento. Para a etapa de treinamento os seguintes passos devem ser feitos:

- Adquirir uma coletânea de imagens para ser o conjunto de treinamento;
- Deixar as imagens no tamanho desejado, cortando somente a face como objeto de interesse;
- Fazer o vetor de imagens ( $I = N \times N$ ) ser um vetor gamma ( $\Gamma = N^2 \times 1$ );
- Calcular a face média do vetor  $\Gamma$  a partir da fórmula  $\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$
- Calcular a face média de cada face do conjunto  $\Gamma$  a partir da fórmula  $\Phi_i = \Gamma_i - \Psi$
- Obter a matriz de covariância  $C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = AA^T$ . Onde  $A = [\Phi_1 \Phi_2 \dots \Phi_M]$
- Autovalores e Autovetores
- Obter os melhores M autovetores
- Manter somente os K melhores autovetores (K com maiores autovalores)

A segunda etapa sendo a de reconhecimento, se utiliza uma face que esta fora do conjunto de treinamento para poder fazer a validação, além de que deve estar com o mesmo tamanho das imagens do conjunto de treino, seguindo os seguintes passos:

- Normalizar a imagem  $\Gamma$  :  $\Phi = \Gamma - \Psi$
- Calcular um conjunto de pesos baseados na imagem de entrada e as M Eigenfaces projetando a imagem de entrada em cada uma das Eigenfaces;
- Determinar se a imagem de entrada é uma face, calculando se a imagem está suficientemente próxima do espaço facial;
- Se for uma face, classificar os padrões de peso para saber se é uma pessoa conhecida ou desconhecida;

Segundo (COIFMAN; WICKERHAUSER, 1995), (PARRAGA, 2002), e este ainda (OPPENHEIM, 2009). Exemplo de ref

**Altura** (*Pitch*)

## 2.2 SISTEMAS EMBARCADOS

Atualmente a tecnologia está cada vez mais popular e acessível, sendo assim possui-se diversos tipos diferentes de tecnologias para os mais diversos fins, sendo assim um conceito para que essas tecnologias possam ser classificadas e separadas para os seus específicos fins, pode-se utilizar o conceito de sistemas embarcados. Os sistemas embarcados são tecnologias que são construídas para devidos fins específicos com hardware específico que acaba não podendo ser reutilizado em outras aplicações que não possuam o mesmo hardware utilizado em seu desenvolvimento original.

Os sistemas embarcados podem normalmente desempenhar processos simples, que não geram nenhum tipo de risco para os usuários (exemplo de calculadoras, controles de videogames, telefones, etc.), contudo também podem ser utilizados para tarefas mais complexas que apresentam certos riscos se não forem projetados com um maior rigor (exemplo controle em aviões, controles industriais, monitoramento de saúde, etc.).

Neste âmbito um sistema desenvolvido para um hardware específico pode ser considerado um sistema embarcado, neste tipo de aplicação um hardware muito utilizado atualmente são as RaspberryPi, que são microprocessadores que rodam sistemas operacionais de diversos tipos.

### 2.2.1 Sons normais

exemplo de tabela

(TURK; PENTLAND, 1991a) (TURK; PENTLAND, 1991b) (PARRAGA, 2002) (OPPENHEIM, 2009) (COIFMAN; WICKERHAUSER, 1995)

Tipo	Intensidade	Descrição
Vesicular	Suave	Formado pela passagem do ar pelo parênquima pulmonar.
Bronquial	Alta	Som traqueal audível na zona de projeção de brônquios de maior calibre.
Broncovesicular	Intermediária	Somam-se as características do som bronquial com o som vesicular.
Traqueal	Muito alta	Som produzido na traqueia pela passagem do ar.

Tabela 2.1: Tipos de sons normais.

#### 2.2.1.1 FORMULAS USADAS

A convolução é uma ...

Caso a entrada neste sistema seja um impulso, o sinal de saída será denominada como resposta impulsiva e representado como  $h[n]$ , na eq. 2.16.

Pra salvar as equações e variáveis que estou utilizando no momento.

$$N = 300 \quad (2.1)$$

$$N^2 = 90000 \quad (2.2)$$

$$M = 196 \quad (2.3)$$

$$K = 50 \quad (2.4)$$

$$I \rightarrow N \times N \quad (2.5)$$

$$\Gamma \rightarrow N^2 \times 1 \quad (2.6)$$

$$\Psi \rightarrow N^2 \times 1 \quad (2.7)$$

$$\Phi_i \rightarrow N^2 \times 1 \quad (2.8)$$

$$C \rightarrow N^2 \times N^2 \quad (2.9)$$

$$A \rightarrow M \times N^2 \quad (2.10)$$

$$C_{mod} \rightarrow M \times M \quad (2.11)$$

$$u \rightarrow M \times N^2 \quad (2.12)$$

$$w_i \rightarrow 1 \times 1 \quad (2.13)$$

$$\Omega_i \rightarrow K \times 1 \quad (2.14)$$

$$\Omega_i = \lceil w_1 \rceil |w_2| \quad (2.15)$$

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n - k] \quad (2.16)$$

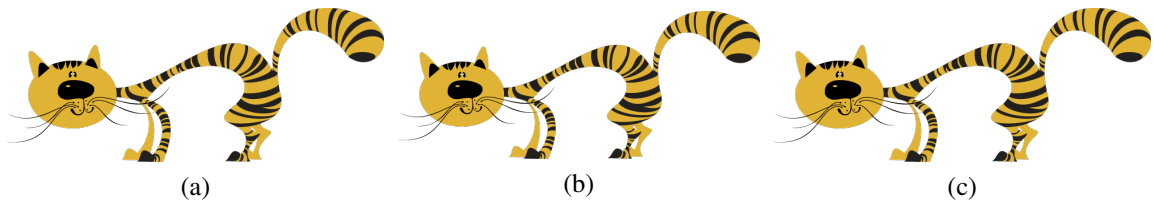


Figura 2.2: Espectros dos sinais: (a) sinal  $X(f)$ , (b) trem de impulsos, (c) sinal  $X(f)$  multiplicado pelo trem de impulsos.

## 3 MATERIAIS E MÉTODOS

Neste capítulo serão abordados os materiais utilizados para a construção e desenvolvimento do presente trabalho, bem como os métodos discutidos anteriormente.

### 3.1 MATERIAIS

A aplicação será desenvolvida utilizando a linguagem de programação Python, sendo ela a linguagem escolhida devido ao seu maior acervo bibliográfico referente as funções necessárias para a o desenvolvimento de presente trabalho.

Para o tratamento e requisições das imagens, será utilizado a biblioteca OpenCV, devido ao seu alto grau de conceituação na literatura atual além de ser de código aberto e com uma ampla disponibilidade de documentação.

Para o armazenamento das informações necessária para gerar as presenças dentro da aplicação, será utilizado o banco de dados não relacional MongoDB, devido a não necessidade de relação entre as informações inseridas nas tabelas do banco criado, assim como a inserção das informações das imagens. Além de possuir conexão com a linguagem escolhida para a criação da aplicação. Para o desenvolvimento da interface gráfica, será utilizado o framework de desenvolvimento gráfico PyQt, gerando uma interface ?amigável? para todos os tipos de usuários, além de disponibilizar mais facilmente as opções de gerência do banco de dados utilizado.

O banco de imagens para o treinamento e o teste dos métodos previamente descritos, são o banco FEI face database, YALE face database e um banco de dados criado manualmente. O banco FEI é um banco de imagens faciais brasileiro que possui 14 imagens de 200 indivíduos, totalizando 2800 imagens. O banco YALE possui 165 imagens de 15 indivíduos, totalizando 2475 imagens.

Os testes serão realizados em uma Raspberry Pi 3, que possui um processador ARMv8 CORTEX A53 QUADCORE com velocidade de operação de 1.2GHz e 1GB de memória RAM. Utilizando o sistema operacional Linux para executar a aplicação desenvolvida.

### 3.2 PYTHON

Python é uma linguagem de programação considerada de alto nível, tendo o seu funcionamento interpretado, onde o programa é executado por um interpretador e após pelo sistema operacional ou processador. A linguagem também é orientada a objetos e funcional. Foi criada em 1991. Possui uma grande comunidade, já que a linguagem possui um modelo de desenvolvimento comunitário, onde várias pessoas podem ajudar a desenvolver novas bibliotecas para que toda a comunidade possa utilizar.



### 3.3 OPENCV

O OpenCV é uma biblioteca de visão computacional e aprendizagem de máquina com código livre, possuindo mais 2500 algoritmos otimizados para uso de todos. Os algoritmos vão desde detectar objetos até extração de modelos 3D. A biblioteca pode ser utilizada nas linguagens de programação Python, C++, Java e MatLab, tendo seus códigos escritos nativamente em C++.

### 3.4 MONGODB

O MongoDB é um banco de dados não relacional que utiliza documentos para armazenar informações, utilizando informações na linguagem JSON. Podendo se conectar com as mais diversas linguagens de programação (como por exemplo: Java, Python, Ruby, C++, etc).

### 3.5 EIGENFACES

Eigenfaces é um classificador de faces utilizado em visão computacional, utiliza auto-vetores para poder fazer a aproximação do valor de rosto médio com o rosto em questão. A criação do conceito de eigenfaces foi criada em 1987 por Sirovich e Kirby, tendo como primeira intenção de uma aproximação de uma representação de imagens em dimensões menores. As eigenfaces podem ser geradas a partir de um processo matemático chamado de Análise de Componente Principal (PCA) em um banco de imagens de faces consideravelmente grande.

### 3.6 METODOLOGIA

Nesta seção serão apresentadas as etapas de desenvolvimento do presente trabalho seguindo o caminho da fig. 3.1.

O desenvolvimento do presente trabalho possui uma construção paralela das funções para que se possa ir integrando cada parte do sistema, afim de chegar no final do desenvolvimento com todas as funções funcionando integralmente, sendo assim o início do desenvolvimento segue a etapa de aquisição das imagens, tendo em paralelo o desenvolvimento de uma interface que seja de simples acesso e o mais claro possível para o usuário final. As imagens serão adquiridas utilizando um algoritmo, contendo a biblioteca OpenCV, sendo executado na Raspberry Pi juntamente com uma câmera (podendo ser um webcam ou uma Raspicam).

Após a obtenção das imagens é necessário fazer a segmentação do rosto encontrado, considerando somente a face (do queixo até início da cabeça), em paralelo será construído e integrado o banco de dados com a interface, já que será necessário analisar como os dados das imagens adquiridas serão armazenadas no banco de dados (para que não seja necessário armazenar nenhum tipo de informação na RaspberryPi, devido ao seu espaço limitado de armazenamento).

Assim que a face estiver segmentada, poderá ser aplicado o classificador Eigenfaces (a segmentação se faz necessária para que o fundo da imagem e qualquer outro objeto não influencie na classificação do indivíduo) para o reconhecimento do indivíduo em questão, como dito anteriormente, em paralelo será testado funções de frequência/presença no sistema, para que o objetivo final do sistema, que seria de por meio do reconhecimento

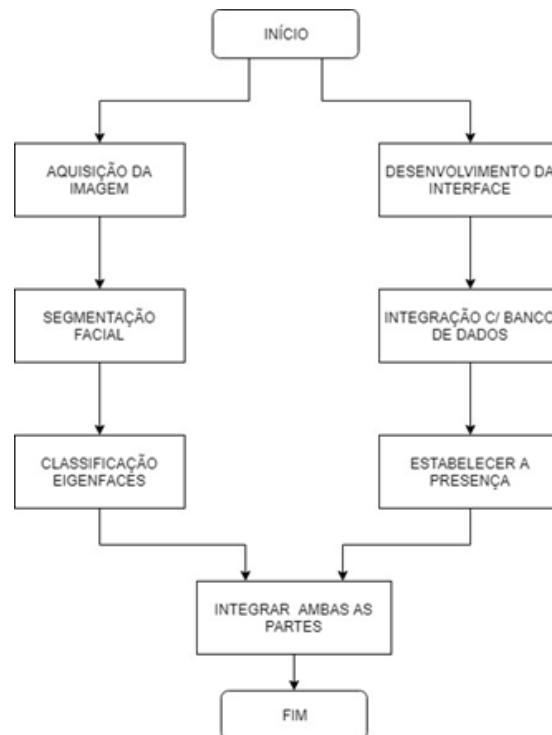


Figura 3.1: Diagrama das etapas de desenvolvimento.

facial a presença do indivíduo, da qual quer ser reconhecido, receberá presença dentro do sistema.

Na sua etapa final, todas as partes desenvolvidas serão integradas em uma versão final, já que ao longo do desenvolvimento do sistema as partes serão testadas para não ter nenhum tipo de problema na integração dos mesmos. O funcionamento do sistema pode ser visto na fig. 3.2.

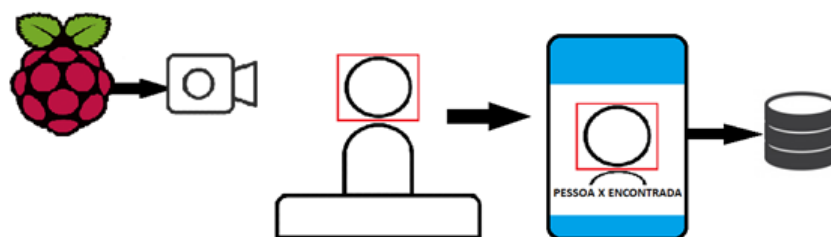


Figura 3.2: Diagrama do sistema.

## 4 RESULTADOS E DISCUSSÕES

Nesta seção serão discutidos os resultados obtidos a partir da aplicação construída e das eigenfaces obtidas.

### 4.1 Obtenção das Eigenfaces

A primeira etapa para o processo de obtenção das eigenfaces é obter um conjunto de imagens, estas imagens serão o grupo de treinamento, após deve-se transformar cada imagem  $I : (N \times N)$  em um vetor gama  $\Gamma : (N^2 \times 1)$ , onde é necessário transformar o vetor em uma dimensão, para reduzir o tamanho dos cálculos as imagens foram dimensionadas no tamanho  $M = 300 \times 300$  e possuem somente um canal em escala de cinza, um exemplo da imagem de treinamento pode ser vista na figura 4.1. As imagens podem ser normalizadas dividindo pelo valor máximo de 255 para facilitar os cálculos.

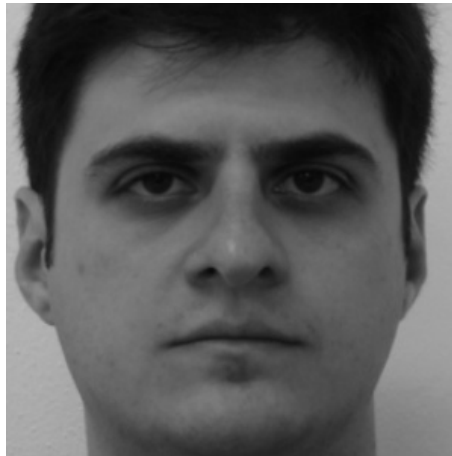


Figura 4.1: Face pertencente ao treino.

Após este processo de achatamento da imagem, deve-se obter a face média do treinamento, sendo uma soma de todas as imagens presentes no treinamento e dividido pela quantidade de imagens, sendo obtida uma imagem  $\Psi$  na equação 4.1.

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (4.1)$$

A imagem gerada é um amalgama de todas as faces de treino, possuindo um pouco de cada característica mais exuberante de cada face, o resultado da junção das faces pode ser notada na figura 4.2.



Figura 4.2: Face média  $\Psi$ .

Cada face  $\Gamma$  pode ser subtraída da face média  $\Psi$  para assim formar um vetor de diferenças  $\Phi$  presente na equação 4.2, sendo estas as diferenças mais distintas de cada face, essa diferença pode ser notada nas figura 4.3.

$$\Phi_i = \Gamma_i - \Psi \quad (4.2)$$

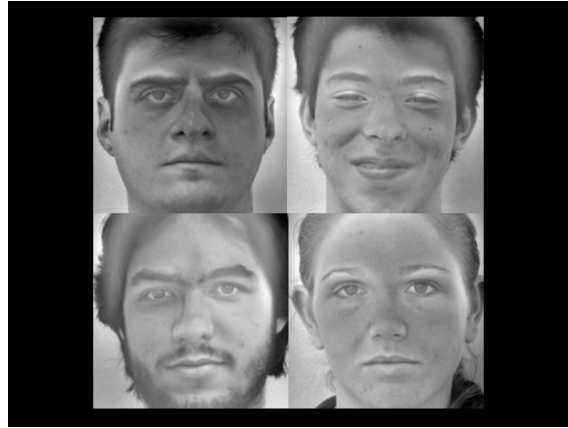


Figura 4.3: Faces  $\Phi$ .

Os autovetores e autovalores podem ser obtidas a partir da matriz de covariância do vetor  $AA^T$  como pode ser visto na equação 4.3, sendo este um vetor de  $\Phi$ , porém esta matriz possui um tamanho que pode ser um desperdício computacional ( $N^2 \times N^2$ ) então é necessário fazer o cálculo a partir da matriz transposta  $A^T$  que é computacionalmente menor ( $M \times M$ ).

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = AA^T \quad (N^2 \times N^2) \quad (4.3)$$

Sendo assim é necessário calcular os autovetores a partir de  $A^T A$  e achar os autovetores de  $AA^T$  a partir da relação da equação 4.4.

$$A^T A v_i = \mu_i v_i \Rightarrow AA^T A v_i = \mu_i A v_i \Rightarrow C A v_i = \mu_i A v_i \Rightarrow \text{onde} \rightarrow u_i = A v_i \quad (4.4)$$

Sendo assim é possível encontrar os autovetores da matriz de covariância  $AA^T$ , sendo estes os maiores autovetores de dos autovetores de  $A^T A$ . Os autovetores podem ser referenciados como as eigenfaces que devem ser redimensionadas para que seja possível ser visualizadas em uma imagem. Alguns exemplos de eigenfaces obtidas podem ser vistas na figura 4.4.



Figura 4.4: Eigenfaces.

Com as eigenfaces geradas é necessário calcular os pesos de cada eigenface nas diferentes faces do treinamento, os pesos calculados para cada face são obtidos utilizando as eigenfaces transpostas vezes a face  $\Phi$  como na equação 4.5.

$$w_j = u_j^T \Phi_i \quad (4.5)$$

Os pesos  $w_j$  são distintos para cada face, esses pesos mais a face média  $\Psi$ , resulta na face original  $\Phi$ , contudo essa reconstrução pode não ocorrer do jeito esperado, já que cada rosto possui uma forma própria mesmo que elas sejam alinhadas. Para representar as faces no "espaço de faces" é necessário reunir cada peso  $w_j$  de cada face em um vetor  $\Omega$  como na equação 4.6

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ w_3^i \\ \vdots \\ w_K^i \end{bmatrix}, \quad i = 1, 2, \dots, M \quad (4.6)$$

## 4.2 Interface

Para a utilização das eigenfaces, foi desenvolvido uma interface utilizando a biblioteca Tkinter. O objetivo da interface é mostrar para o usuário que a sua face esta sendo capturada e sendo calculado se a sua face será reconhecida, devido ao fato de que as pessoas que serão submetidas ao sistema tem o interesse de serem reconhecidas.

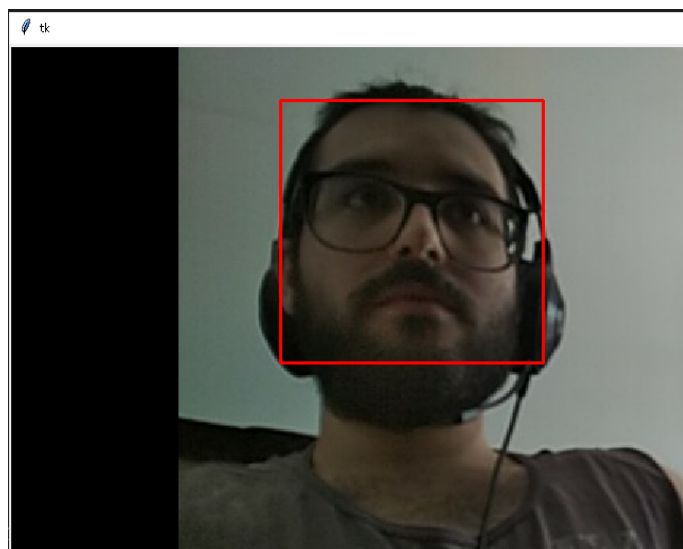


Figura 4.5: Interface do sistema.

## **5 CONCLUSÃO**

## **6 TRABALHOS FUTUROS**



## REFERÊNCIAS

COIFMAN, R. R.; WICKERHAUSER, M. V. LOL Experiments with Adapted Wavelet De-Noising for Medical Signals and Images. In: TIME FREQUENCY AND WAVELETS IN BIOMEDICAL SIGNAL PROCESSING, SER. IEEE PRESS SERIES IN BIOMEDICAL ENGINEERING, 1995. **Anais...** [S.l.: s.n.], 1995. p.323–346.

OPPENHEIM, A. V. **Discrete-Time Signal Processing**. F. Watts, London, England: Prentice Hall, 2009.

PARRAGA, A. **Aplicação da transformada Wavelet Packet na análise e classificação de sinais de vozes patológicas**. [S.l.]: Dissertação UFRGS, 2002.

TURK, M.; PENTLAND, A. P. **Face Recognition Using Eigenfaces**. 1991.

TURK, M.; PENTLAND, A. P. **Eigenfaces for Recognition**. 1991.