

UNIVERSIDADE ESTADUAL DO RIO GRANDE SUL
ENGENHARIA DE COMPUTAÇÃO

ALLEFF DYMYTRY PEREIRA DE DEUS

**RECONHECIMENTO FACIAL PARA
ATRIBUIÇÃO DE PRESENÇA EM SALA
DE AULA**

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

Prof.^a. Dr.^a. Letícia Vieira Guimarães
Orientador

Guaíba, Janeiro de 2021

FOLHA DE APROVAÇÃO

Monografia sob o título "*Reconhecimento Facial para Atribuição de Presença em Sala de Aula*", defendida por Alleff Dymytry Pereira de Deus e aprovada em 21 de Janeiro de 2021, em Guaíba, estado do Rio Grande do Sul, pela banca examinadora constituída pelos professores:

Prof^a. Dr^a. Letícia Vieira Guimarães
Orientadora

Prof^a. Dr^a. Adriane Parraga

Prof. Dr. João Leonardo Fragoso

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

AGRADECIMENTOS

Aos meus familiares por possibilitar que eu pudesse realizar esta etapa dentro de minha vida, podendo dar o melhor de mim do início ao fim. Sempre ajudando nos momentos difíceis e também nos momentos de alegrias.

À minha noiva, por poder dividir todos os momentos que foram a graduação, sempre me ajudando em situações adversas, sempre aconselhando e dividindo o conhecimento obtido. Ajudando a escalar esta montanha que não é fácil de chegar ao fim.

Aos meus professores, que possibilitaram o conhecimento e o crescimento dos ideais, sempre aconselhando e mostrando os melhores trilhos para seguir nesta jornada.

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

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE SÍMBOLOS	8
LISTA DE FIGURAS	9
LISTA DE TABELAS	10
RESUMO	11
ABSTRACT	12
1 INTRODUÇÃO	13
1.0.1 Objetivo Geral	13
1.0.2 Objetivos Específicos	14
1.0.3 Motivação	14
2 FUNDAMENTOS TEÓRICOS	15
2.1 PROCESSAMENTO DIGITAL DE IMAGENS	15
2.1.1 Imagem digital	15
2.1.2 Processamento Digital	15
2.2 CARACTERÍSTICAS DO TIPO HAAR	16
2.3 ANÁLISE DE COMPONENTES PRINCIPAIS	16
2.4 EIGENFACES	17
2.4.1 Reconhecimento Utilizando Eigenfaces	18
2.5 SISTEMAS EMBARCADOS	18
3 MATERIAIS E MÉTODOS	20
3.1 Interface	22
3.1.1 Modo de Supervisão	22
3.1.2 Modo de Atribuição de Presença do Aluno	23
3.2 Aquisição da Imagem	23
3.3 Região de Interesse (ROI)	24
3.4 Eigenfaces	25
3.5 Reconhecimento	28
3.6 Banco de Dados	29
3.7 Resultado do reconhecimento	31
3.8 Integração do Sistema	31

4	RESULTADOS E EXPERIMENTOS	33
4.1	Descrição dos Testes com as Eigenfaces	33
4.1.1	Percentual de Acerto no Reconhecimento	33
4.1.2	Determinação do valor de K	34
4.1.3	Efeito do desvio padrão nos pesos Ω	35
4.1.4	Comparação do desempenho entre os parâmetros de reconhecimento (Erro mínimo de Ω x Erro mínimo de Φ)	35
4.2	Teste de tempo de resposta do Banco de Dados	43
5	CONCLUSÃO E TRABALHOS FUTUROS	45
	REFERÊNCIAS	47

LISTA DE ABREVIATURAS E SIGLAS

PCA	Principal Component Analysis
ROI	Region Object Interest
RGB	Red Green Blue

LISTA DE SÍMBOLOS

Γ	Vetor unidimensional da Face
Ψ	Face Média do conjunto de faces
Φ	Face com principais componentes ressaltadas
Ω	Conjunto de pesos de cada Eigenface para determinada face Φ
e_{Ω}	Erro das diferenças dos pesos Ω
e_{Φ}	Erro das diferenças das faces Φ
θ_{Ω}	Valor de limiar dos pesos Ω
θ_{Φ}	Valor de limiar das faces Φ

LISTA DE FIGURAS

2.1	Exemplos de região retangular para análise de características do tipo Haar.	16
3.1	Diagrama da divisão do sistema.	21
3.2	ROI da face.	25
3.3	Faces sem tratamento pertencentes ao treino.	25
3.4	Face tratada pertencente ao treino.	26
3.5	Face média Ψ	26
3.6	Faces Φ	27
3.7	Eigenfaces.	27
3.8	Diagrama do sistema.	32
4.1	Face de teste, Face reconhecida do banco de dados: (a) Indivíduo reconhecido, (b) Indivíduo não reconhecido	34
4.2	Gráfico K x Acertos.	34
4.3	Gráfico K x Erros.	35
4.4	Desvio Padrão dos Omegas projetados no espaço de faces.	35
4.5	Desvio Padrão dos Omegas no espaço de faces.	36
4.6	Faces reconhecidas do treinamento utilizando Erro $\Phi = 30$	37
4.7	Faces reconhecidas do treinamento utilizando Erro $\Phi = 31$	37
4.8	Faces reconhecidas do treinamento utilizando Erro $\Phi = 32$	38
4.9	Faces reconhecidas do treinamento utilizando Erro $\Phi = 33$	38
4.10	Faces reconhecidas do treinamento utilizando Erro $\Phi = 34$	39
4.11	Faces reconhecidas do treinamento utilizando Erro $\Phi = 35$	39
4.12	Faces reconhecidas de fora do treinamento utilizando Erro $\Phi = 30$	40
4.13	Faces reconhecidas de fora do treinamento utilizando Erro $\Phi = 31$	40
4.14	Faces reconhecidas de fora do treinamento utilizando Erro $\Phi = 32$	41
4.15	Faces reconhecidas de fora do treinamento utilizando Erro $\Phi = 33$	41
4.16	Faces reconhecidas de fora do treinamento utilizando Erro $\Phi = 34$	42
4.17	Faces reconhecidas de fora do treinamento utilizando Erro $\Phi = 35$	42
4.18	Armazenamento das imagens a partir dos Bancos de Dados.	44

LISTA DE TABELAS

4.1	Falsos positivos e negativos	43
4.2	Tempo de recuperação das imagens	44

RESUMO

A popularização das câmeras digitais, microprocessadores e aplicações de internet, proporcionaram o desenvolvimento de algumas novas técnicas para a utilização do reconhecimento facial. Este trabalho apresenta o desenvolvimento e implementação de um sistema embarcado que reconhece um indivíduo e atribui a sua presença em sala de aula ou em evento, priorizando a portabilidade e facilidade de acesso ao sistema. Assim é sugerido a utilização da Raspberry Pi para embarcar o sistema proposto, devido a sua portabilidade e possibilidade de alimentação por baterias externas. A construção de um sistema de reconhecimento facial automático para o controle de presença de aluno em sala de aula se faz necessário principalmente pelo tempo gasto durante o processo de chamada manual. Além disso a maioria das instituições de ensino não disponibiliza ao professor uma foto junto ao nome do aluno na ata de presença, assim não havendo um meio de conferir se de fato é o aluno em questão que está presente. Mesmo utilizando alguns recursos tecnológicos como formulários online, ainda precisam que sejam encaminhados e preenchidos manualmente para poder se ter a validação. A técnica de reconhecimento facial implementada é baseada em características denominadas de Eigenfaces composta por autovetores e autovalores das imagens das faces de entrada. Os testes realizados mostram uma taxa de acerto no reconhecimento das faces 10% maior do que a mencionada na literatura de referência. A faixa de percentual de acerto do sistema proposto foi de 76% a 88% de acerto do conjunto de faces utilizadas, onde as imagens das faces utilizadas possuíam modificações mínimas nas expressões faciais. Em síntese o indivíduo que quer ser reconhecido pelo sistema não irá apresentar adversidades para o seu reconhecimento acrescentando elementos estranhos à face, já que ele quer a aprovação no sistema. Portanto a taxa de acerto obtida nos testes pode ser considerada adequada à aplicação proposta.

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

Facial Recognition for Assignment on Presence in Classroom

ABSTRACT

The popularization of digital cameras, microprocessors, and internet applications provided the development of some new techniques for the use of facial recognition. This work presents the development and implementation of an embedded system that recognizes an individual and assigns its presence in the classroom or an event, prioritizing portability and ease of access to the system. Thus is suggested to the use of Raspberry Pi board the proposed system, due to its portability and possibility of powering by external batteries. The construction of an automatic facial recognition system to control the presence of students in class is necessary mainly for the time spent during the manual call process. Besides, most educational institutions do not provide the teacher with a photo next to the student's name in the attendance record, so there is no way to check if it is the student in question who is present. Even using some technological resources such as online forms, they still need to be forwarded and filled out manually to have the validation. The implemented facial recognition technique is based on features called Eigenfaces composed of eigenvectors and eigenvalues of the images of the input faces. The tests performed to show a hit rate in the recognition of the faces 10% higher than the one mentioned in the reference literature. The proposed system's hit percentage range was from 76% to 88% percent of the faces used, where the images of the faces used had minimal modifications in the facial expressions. In summary, the individual who wants to be recognized by the system will not present adversities for his or her recognition by adding strange elements to the face, since he or she wants the approval in the system. Therefore, the hit rate obtained in the tests can be considered adequate for the proposed application.

Keywords: Eigenfaces, Facial Recognition, Haar Like Features.

1 INTRODUÇÃO

Nos dias atuais pode-se notar que a biometria é algo amplamente utilizado em variados aspectos do cotidiano da sociedade. Dentre as funções que a biometria é aplicada temos: pagamento de contas, liberação para áreas restritas, verificação em e-mails, para direção de um veículo, etc.

A biometria é fundamentada na utilização da medição de características humanas de forma analógica transformando para o mundo digital, um exemplo de como a utilização da biometria tem sido ampliada é o seu uso no Brasil para operações eleitorais e cada vez mais vem sendo utilizadas tecnologias mais avançadas para os mais diferentes fins, da mesma forma é percebido um grande avanço nas técnicas de reconhecimento facial como biometria.

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% até 30% por ano na área. O reconhecimento facial tem sido debatido para muitos usos, sendo o uso para a segurança pública um dos usos mais comuns.

Durante a pandemia de 2020, os sistemas de reconhecimento facial acabaram ganhando uma popularização devido a necessidade de se reconhecer um indivíduo sem a necessidade da interação física com o sistema. Segundo a revista Digital Security, ver (SECURITY, 2020), no Paraná em menos de cinco meses após o início da pandemia, houve um crescimento de mais de 20 mil pessoas impactadas na utilização do reconhecimento facial, tanto para o uso na entrada de condomínios, quanto de empresas e outros setores. Sendo assim 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.

1.0.1 Objetivo Geral

O objetivo geral do presente trabalho é desenvolver um sistema embarcado que reconheça um indivíduo e atribua a sua presença em sala de aula ou em eventos, priorizando a portabilidade e facilidade de acesso ao sistema.

A aplicação de reconhecimento facial para atribuição de presença em sala de aula, torna possível substituir o método tradicional de folha de chamada com preenchimento manual por um sistema automático integrado em um sistema embarcado. E a utilização do projeto proposto não precisa ficar somente nas salas de aula, as aplicações podem ser utilizadas para conferir a presença de um indivíduo nos mais diferentes eventos, onde o mesmo quer ser reconhecido para obter algum certificado que esteve presente no momento

em que foi reconhecido.

Os valores da taxa de acerto obtidos na pesquisa de (TURK; PENTLAND, 1991) são divididos em dois testes, o primeiro com todas as faces sendo classificadas como reconhecidas obteve uma taxa 64% em relação a variação do tamanho da imagem da face original e a de teste, o segundo teste somente as imagens das faces que possuíam projeções próximas das conhecidas seriam atribuídas como reconhecidas obteve uma taxa de 74%. Assim com os valores da taxa de acerto obtido por (TURK; PENTLAND, 1991), o sistema proposto tem por taxa de acerto esperada de 65% das imagens de faces utilizadas para compor o conjunto de treinamento do sistema.

1.0.2 Objetivos Específicos

Os objetivos específicos do presente trabalho são: (1) implementar técnicas de reconhecimento de imagens por autovetores, autovalores e PCA, (2) implementar o sistema em um ambiente de ensino, mais especificamente na UERGS na unidade de Guaíba, e (3) a produção de um banco de imagens de faces com os alunos matriculados na UERGS na unidade de Guaíba.

1.0.3 Motivação

A popularização das câmeras digitais, microprocessadores e aplicações de internet, algumas novas técnicas foram sendo desenvolvidas para a utilização de reconhecimento facial. Algumas das técnicas utilizadas atualmente são: Eigenfaces, Fisherfaces, Kernel Direct Discriminant Analysis, K Nearest Neighbors, Local Binary Pattern e etc. Técnicas que incorporam reconhecimento de padrões, redes neurais e inteligência artificial, também tem sido amplamente utilizadas.

A construção de um sistema de reconhecimento facial automático para o controle de presença de aluno em sala de aula se faz necessário principalmente pelo tempo gasto durante o processo de chamada manual. Além disso a maioria das instituições de ensino não disponibiliza ao professor uma foto junto ao nome do aluno na ata de presença, então não tem como conferir se de fato é o aluno em questão que está presente.

E um tempo de integração e avanço tecnológico, utilizar chamadas impressas para marcar se o aluno estava ou não em uma aula ou em um determinado evento se torna arcaico e impreciso. Mesmo utilizando alguns recursos tecnológicos como formulários online, ainda precisam que seja encaminhado e preenchidos manualmente para poder se ter a validação.

2 FUNDAMENTOS TEÓRICOS

Neste capítulo são abordados os conhecimentos necessários para o entendimento do presente trabalho, bem como seus métodos de desenvolvimento e funcionalidades específicas.

2.1 PROCESSAMENTO DIGITAL DE IMAGENS

Para poder abordar como são feitos os processos sobre as imagens, primeiramente é necessário se fundamentar o que são imagens em um computador.

2.1.1 Imagem digital

Uma imagem pode ser descrita como uma função bidimensional, com dimensões de tamanhos iguais ou diferentes. As coordenadas da imagem formam um plano e sua amplitude pode ser descrita como a intensidade do conjunto de coordenadas. O valor da função que descreve a imagem é resultado de processos físicos relacionados à energia irradiada pela fonte. Uma imagem pode ser captada a partir de sensores que conseguem traduzir a energia irradiada pela fonte em valor de tensão. Segundo (GONZALEZ, 2010), um dos sensores mais conhecidos para tal utilização, são os fotodiodos, sendo estes sensores construídos com materiais semicondutores que possuem uma saída de tensão proporcional a intensidade luminosa. Entretanto, a imagem digital tem valores dos elementos de coordenadas e amplitude finitas. Os elementos que compõem uma imagem digital são chamados de *Pixels*.

Uma imagem digital, constituída de vários *Pixels*, pode possuir mais de um plano, constituindo um conjunto de planos que pode ser chamado de canais. Os canais em uma imagem podem possuir diferentes representações de cores específicas, sendo muito utilizados os padrões de cores RGB (Vermelho, Verde e Azul - Red, Green e Blue), YCM (Amarelo, Ciano e Magenta - Yellow, Cyan e Magenta) e entre outros, também é possível utilizar o canal com sua cor em uma escala de cinza.

2.1.2 Processamento Digital

A imagem digital, pode ser processada por um sistema computacional de forma que pode-se abordar que o processamento digital de imagens sendo a área que utiliza imagens como entrada principal com diversos propósitos, por exemplo, realçar certas características de objetos na imagem. O processamento de imagem apresenta como saída uma nova imagem. Contudo alguns autores consideram certas operações como sendo parte da visão computacional, já que o processamento é apontado como uma função que recebe uma imagem e retorna uma imagem. Todavia, funções como obtenção e classificação de infor-

mações nas imagens são consideradas parte da visão computacional e análise de imagens. Segundo (GONZALEZ, 2010), não existem limites claros onde se pode considerar que uma operação faz parte de qual linha de pesquisa, o que se pode traçar é um paradigma que utiliza três níveis de operações, sendo eles: o nível de processo baixo, médio e alto. O nível de processamento baixo é constituído de pré-processamentos e as mais diversas operações primitivas que se pode fazer com uma imagem. O nível de processo médio constitui-se com tarefas de segmentação de ROIs, descrição para a posterior utilização na classificação dos objetos segmentados. E o nível de processo alto seria a área que atribui um sentido lógico e real para as informações adquiridas pelos dois níveis inferiores.

2.2 CARACTERÍSTICAS DO TIPO HAAR

As características do tipo Haar ou (Haar Like Features) são características baseadas nos Haar wavelets desenvolvidas por Alfred Haar, onde transforma um sinal em uma representação mais simples para certos procedimentos de análise. Primeiramente desenvolvido por (VIOLA; JONES, 2001), são características utilizadas para o reconhecimento de objetos (neste caso o reconhecimento de faces). As características utilizadas para o reconhecimento se baseiam no cálculo da diferença entre os pixels dentro de uma região retangular, essas características são divididas em três: 1) diferença entre dois retângulos, tanto verticais como horizontais; 2) diferença entre três retângulos, considerando a soma dos retângulos das bordas com a diferença do retângulo central e; 3) diferença diagonal de pares de retângulos, um exemplo dessas características pode ser notado na Figura 2.1.

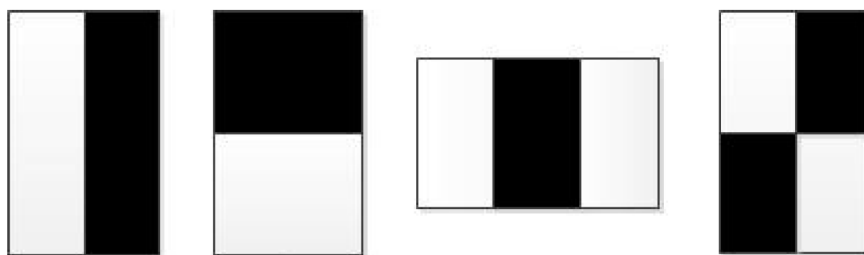


Figura 2.1: Exemplos de região retangular para análise de características do tipo Haar.

2.3 ANÁLISE DE COMPONENTES PRINCIPAIS

A Análise de Componentes Principais (Principal Component Analysis - PCA) é uma técnica matemática utilizada para a análise de dados de para sistemas com múltiplas variáveis. Primeiramente concebido por Karl Pearson em 1901 e posteriormente desenvolvida por Harold Hotelling em 1930. O PCA não era utilizado tão comumente até os avanços na área da computação. Este método se baseia na utilização de princípios matemáticos para transformar um número de variáveis correlacionais em quantidades menores de variáveis, sendo estas as principais componentes das variáveis. A principal transformação é a redução da dimensionalidade do conjunto de dados, fazendo com que sejam ressaltadas as similaridades e diferenças dentro do conjunto de dados. Segundo (MISHRA et al., 2017) os autovetores e autovalores são os princípios fundamentais para a determinação dos PCAs, decorrido da decomposição da matriz de covariância. O processo de obtenção dos PCAs pode ser feito através dos seguintes passos:

- **Subtração da Média**

Para que o cálculo do PCA funcione devidamente, é necessário realizar a subtração do conjunto dos dados de sua média em ambos os eixos, para que possa ser produzido uma média dos dados de valor zero;

- **Cálculo da Matriz de Covariância**

É necessário obter a matriz de covariância do conjunto de dados que já foram subtraídos da média, podendo se obter o valor de covariância através da Eq. 2.1, onde x_i e y_i são pontos de um vetor de tamanho n e x' e y' são os valores médios de cada vetor. A matriz de covariância é a junção da covariância de cada objeto das dimensões das variáveis, este cálculo pode ser notado na Eq. 2.2, onde x e y são coordenadas dos vetores X e Y ;

$$cov(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - x')(y_i - y') \quad (2.1)$$

$$cov(X, Y) = \begin{bmatrix} cov(x_1, y_1) & cov(x_1, y_2) & \cdots & cov(x_1, y_n) \\ cov(x_2, y_1) & cov(x_2, y_2) & \cdots & cov(x_2, y_n) \\ \vdots & \vdots & \cdots & \vdots \\ cov(x_n, y_1) & cov(x_n, y_2) & \cdots & cov(x_n, y_n) \end{bmatrix} \quad (2.2)$$

- **Cálculo dos autovetores e autovalores**

Existem várias formas de se calcular os autovetores e autovalores, uma delas é descrita pela Eq.2.3.

$$(A - \lambda I)u = 0 \quad (2.3)$$

Onde 'A' é uma matriz conhecida, ' λ ' é o autovalor de 'A', 'u' são os autovalores de 'A' e 'I' é a matriz identidade;

- **Escolher as componentes**

Deve-se escolher os maiores autovalores dos quais serão obtidos os maiores autovetores como consequência, sendo estes os valores que mais representam as componentes principais do conjunto de dados.

Ainda, segundo (MISHRA et al., 2017), o PCA pode ser considerado como a projeção do conjunto de dados em uma direção do espaço onde os dados têm uma grande variação, onde a direção é dada pelos autovetores da matriz de covariância correspondendo aos maiores autovalores.

2.4 EIGENFACES

O eigenface é um dos modelos de representação das faces utilizados em algoritmos de reconhecimento facial. O Eigenfaces consiste na obtenção de autovetores que melhor representam (uma representação matemática generalizada de uma face) a face de cada indivíduo. A utilização do termo *Eigenface* foi apresentado por (TURK; PENTLAND, 1991). Neste artigo os autores definem "faces fantasmas" que possuem os maiores autovetores e autovalores do PCA do conjunto de dados. Segundo (TURK; PENTLAND, 1991)

a ideia de utilizar eigenfaces foi motivado na pesquisa de (SIROVICH; KIRBY, 1987), que utilizavam as melhores coordenadas para representar uma face, sendo denominado de *Eigenpictures*. Com isso podia-se reconhecer uma face com uma pequena componente (SIROVICH; KIRBY, 1987), utilizava-se a parte dos olhos para se fazer o reconhecimento, contudo, a ideia de utilizar uma pequena quantidade de características diferentes para reconhecer cada indivíduo persistiu, formando (produzindo) assim uma Eigenface. Desta forma eigenfaces são produzidas multiplicando um conjunto de PCAs da face de cada indivíduo pelos maiores autovetores da matriz de covariância.

2.4.1 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;
- Transformar o vetor de imagens I ($I = N \times N$) em um vetor gamma Γ ($\Gamma = N^2 \times 1$);
- Calcular a face média Ψ do vetor Γ ;
- Obter a matriz de covariância C ;
- Obter os autovalores e autovetores da matriz de Covariância C ;
- Manter somente os K melhores autovetores (K com maiores autovalores);
- Calcular os pesos Ω de cada face Φ pelas eigenfaces.

A segunda etapa sendo a de reconhecimento, se utiliza uma face que está 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 de teste $\Phi = \Gamma - \Psi$
- Calcular pesos Ω da imagem de teste pelas eigenfaces do conjunto de treinamento;
- Determinar se a imagem de entrada é uma face pertencente ao conjunto de treinamento pela distância Euclidiana;

2.5 SISTEMAS EMBARCADOS

Atualmente as tecnologias de microeletrônica e de computação estão cada vez mais com maior capacidade, além de tornarem-se mais populares e acessíveis. Sendo assim encontram-se diversos tipos diferentes de tecnologias para os mais diversos fins. Os sistemas embarcados são tecnologias que são construídas para devidos fins específicos com hardware (circuitos e componentes eletrônicos) 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 Raspberry Pi (RASPBerryPI, 2012), que são microprocessadores que rodam sistemas operacionais de diversos tipos.

3 MATERIAIS E MÉTODOS

Neste capítulo são abordados os materiais e os métodos utilizados para a construção e desenvolvimento do presente trabalho, bem como as bibliotecas necessárias para o correto funcionamento do sistema.

Todas as etapas necessárias para o funcionamento do presente trabalho foram desenvolvidas utilizando a linguagem de programação Python, ver (PYTHON, 2001) que apresenta as seguintes vantagens com a sua utilização: uma linguagem de programação considerada de alto nível, possui o funcionamento interpretado, orientada à objetos e funcional.

O sistema proposto foi desenvolvido e testado em uma máquina com um processador AMD Ryzen 5 3400G de 3,7GHz, memória RAM de 16GB e uma placa de vídeo Radeon RX 590 de 8GB de VRAM.

Todavia, a aplicação final será embarcada em uma Raspberry Pi 3, sendo este um microprocessador com um tamanho consideravelmente pequeno, podendo ser utilizado para as mais diversas funções, possuindo um sistema baseado em Linux como seu sistema operacional, 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.

O sistema proposto segue o fluxo em relação as suas funcionalidades, este fluxo é apresentado na Figura 3.1. O diagrama mostra uma divisão em relação as funções dos processos utilizados pelos dois modos de operação presentes no sistema: supervisão e atribuição de presença.

Para o entendimento do fluxo do sistema, um breve resumo de cada função pode ser elencado. Todavia a descrição de cada funcionalidade do fluxo da Figura 3.1 é elucidada dentro deste capítulo.

- **INTERFACE:** A interface se refere as janelas do sistema, das quais podem ser utilizadas pelos usuários;
- **MODO:** O modo de operação do sistema se refere a separação das atribuições presentes dentro do sistema;
- **ATRIBUIÇÃO DE PRESENÇA DO ALUNO:** O modo de atribuição de presença do aluno se refere ao conjunto de funções necessárias para que a presença seja atribuída ao aluno reconhecido pelo sistema;
- **SUPERVISÃO:** O modo de supervisão se refere as funções específicas do supervisor, que possuem um nível mais elevado de segurança dentro do sistema;

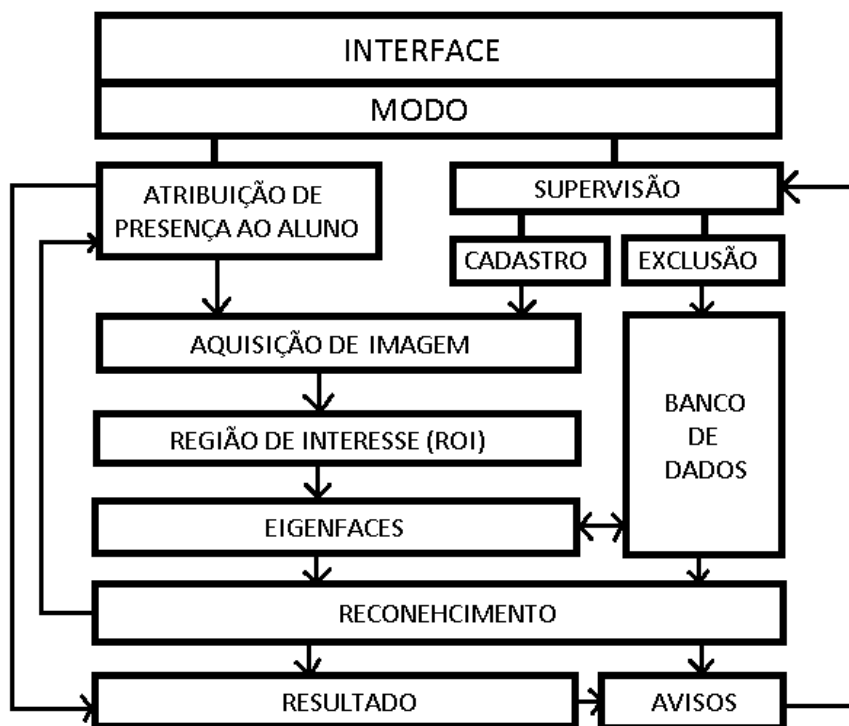


Figura 3.1: Diagrama da divisão do sistema.

- **CADASTRO:** O processo de cadastro se refere a inclusão de novos usuários no sistema, sendo realizado pelo supervisor do sistema;
- **EXCLUSÃO:** O processo de exclusão se refere a retirada de um usuário específico do sistema, esta operação retira todas as informações provenientes do usuário do banco de dados;
- **AQUISIÇÃO DE IMAGEM:** 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.);
- **REGIÃO DE INTERESSE (ROI):** O processo de determinação de região de interesse tem por funcionamento encontrar uma face na imagem adquirida, segmentando para o sistema somente a face localizada;
- **EIGENFACES:** O processo de eigenfaces se refere a utilização e obtenção pelo sistema, onde é necessário um conjunto de imagens para que uma eigenface possa ser produzida;
- **RECONHECIMENTO:** O processo de reconhecimento realiza a interação da imagem da face do usuário com as eigenfaces do sistema.
- **RESULTADO:** O processo de resultados se refere a conclusão do reconhecimento, onde são geradas as informações necessárias para atribuição da presença ou aviso para o supervisor, sendo este o processo que finaliza o fluxo do modo de atribuição de presença do aluno;

- **AVISOS:** O processo de aviso se refere a interface acessada pelo supervisor para avaliar os usuários que não tiveram seus rostos reconhecidos pelo sistema;
- **BANCO DE DADOS:** O banco de dados são todas as informações armazenadas pelo sistema, seja as informações de aula, nome, evento, hora, quanto as informações referentes as faces, eigenfaces e informações necessárias para os cálculos do reconhecimento.

3.1 Interface

A interface do sistema foi desenvolvida utilizando a biblioteca Tkinter, sendo esta uma das bibliotecas padrões para desenvolvimento de interfaces, possui compatibilidade com a maioria dos sistemas operacionais Unix e Windows. A interface do sistema opera em um dos dois modos: Atribuição de presença do aluno ou Supervisão.

3.1.1 Modo de Supervisão

O modo de operação de supervisão possui o intuito de concentrar todas as funções que devem ser realizadas pelo supervisor do sistema, com isso as funções com nível maior de segurança somente estão liberadas para este modo de operação. O modo de Supervisão apresenta as seguintes funções:

- Cadastro de usuários

A função de cadastro de usuários abre uma janela com a função específica de adição de novos usuários ao banco de dados do sistema, onde é necessário que as informações do usuário sejam inseridas no banco de dados bem como a imagem de sua face através de uma janela específica para o supervisor.

A janela de cadastro mostra a imagem capturada pela webcam e ao reconhecer alguma face nesta imagem o sistema mostra ao supervisor a localização da região onde a face se encontra, como pode ser visto na Figura 3.2.

Devido ao cadastro de uma nova face ao banco de faces, se faz necessário recalculer a face Ψ juntamente com os pesos Ω e as eigenfaces do conjunto de treinamento, o conjunto de treinamento é composto de todas as faces do sistema, onde é necessário sempre fazer os cálculos previamente para a utilização no reconhecimento. Após os cálculos é necessário atualizar as informações no banco de dados e no banco de faces.

- Cadastro de aulas e eventos

Para que a presença de um indivíduo seja registrada no sistema, é necessário que seja vinculado a uma sala de aula, disciplina ou evento previamente, sendo assim a função de cadastro de aula/eventos necessita de informações como nome, data, quantidade de pessoas e etc, para que posteriormente seja utilizada no sistema. Nesta função é aberta uma janela onde o supervisor pode colocar todas as informações necessárias para o cadastro de aula ou evento.

- Exclusão de usuários do sistema

Para que o sistema não fique sobrecarregado com imagens que já não são mais necessárias para a utilização, o supervisor pode, se julgar necessário, retirar uma face do sistema, procurando o nome do usuário e excluindo o mesmo, assim o sistema irá realizar a exclusão de seus dados bem como a sua face do banco de faces. Após a exclusão de uma face do sistema, será recalculado a face Ψ , os pesos Ω e as eigenfaces.

- Verificação dos avisos

Durante a execução das funções do sistema, alguns avisos decorrentes do não reconhecimento de um indivíduo pelo sistema podem ser gerados para que o supervisor possa cadastrar como um novo usuário ou editar a face do usuário. A interface de avisos é uma janela onde é possível ser realizado o acesso aos avisos gerados pelo sistema, com o nome do indivíduo, o horário e a aula ou evento.

- Inicialização do modo Atribuição de Presença

O modo de atribuição de presença deve ser inicializado pelo supervisor, sendo assim ele deve acessar a janela onde deve-se escolher a aula ou evento no qual as presenças serão computadas. Assim o modo de Atribuição de presença será executado até que o supervisor finalize a aula ou evento.

3.1.2 Modo de Atribuição de Presença do Aluno

No modo de operação de atribuição de presença do aluno, fica concentrado as funções que serão utilizadas para a aquisição e reconhecimento da face do aluno. Sendo assim este modo de operação fica em execução contínua assim que for iniciado pelo supervisor, gerando os resultados do reconhecimento e computando se o aluno foi ou não reconhecido, a menos que o modo de supervisão seja executado. O sistema executa as funções conforme o fluxo apresentado na Figura 3.1. O fim da execução deste modo ocorre quando o supervisor finaliza a aula ou evento.

- Inserção de informações

Durante o modo de atribuição de presença do aluno, é possível que sua face não seja reconhecida. Sendo assim uma janela é exibida para que o aluno possa inserir manualmente o seu nome, e com isso gerando a sua presença na aula ou evento. Quando esta janela é finalizada, um aviso com o nome e imagem adquirida no momento é gerado para o modo de supervisão, para que seja acessado pelo supervisor posteriormente.

3.2 Aquisição da Imagem

A aquisição da imagem dentro do sistema é realizada a partir de uma webcam que esteja conectada ao sistema, sendo necessário ser feito o acesso da webcam por meio da biblioteca OpenCV (OPENCV, 2021).

O OpenCV é uma biblioteca de visão computacional e aprendizagem de máquina com código livre, possuindo mais 2500 funções otimizadas para uso gratuito. Dentre as funções disponíveis existem algoritmos que vão desde a detecção de objetos até a 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++. Para a utilização da webcam pelo sistema, primeiramente deve-se configurar a mesma utilizando o OpenCV, colocando seus parâmetros iniciais, como por exemplo: tamanho da imagem, quantidade de canais, opção de cor, modo de captura, etc. Sendo assim, os parâmetros utilizados foram: Altura - 480 pixels e Largura - 640 pixels, o parâmetro de cor foi utilizado o RGB.

A aquisição é realizada em ambos os modos de operação, contudo possuem suas peculiaridades específicas, sendo elas:

- Aquisição no modo Supervisão

A aquisição da imagem durante a supervisão do sistema ocorre de forma a se obter imagens para cadastrar um novo aluno ou alterar uma imagem já existente. Sendo assim o supervisor do sistema pode adquirir uma imagem quando achar necessário, durante a aquisição o processo de salvar uma imagem só pode ser feito se o sistema detectar uma face, como na Figura 3.2, caso a face não seja detectada ou mais de uma face seja detectada o sistema não permite que seja adquirida uma imagem, assim evitando falhas no sistema.

- Aquisição no modo de Atribuição de Presença

A aquisição da imagem durante o processo de atribuição da presença ocorre automaticamente assim que uma face é detectada. Em seguida a imagem da mesma é disponibilizada para o sistema, sem a intervenção de um usuário. A imagem adquirida é apresentada na janela da interface por um período de dez segundos para que o indivíduo possa ver que seu rosto foi adquirido pelo sistema.

3.3 Região de Interesse (ROI)

A região de interesse na imagem é segmentada do restante através de detecção e a segmentação da face na imagem adquirida pelo sistema. Desta forma os processos relacionados ao reconhecimento da face tornam-se mais rápidos e o número de falhas no reconhecimento diminui. A face é detectada com as características do tipo Haar para faces frontais.

A segmentação das faces ocorre em ambos os modos de operação do sistema:

- ROI do modo Supervisão

A determinação da ROI durante o modo de supervisão acontece sobre as imagens adquiridas previamente pelo supervisor do sistema, estas imagens são processadas antes da determinação da ROI para possuírem somente um canal em escala de cinza, sendo processadas utilizando as características Haar para retirar somente a face frontal sendo esta a região de interesse.

- ROI do modo Atribuição de Presença

A procura do ROI durante o modo de atribuição de presença do aluno, acontece na imagem adquirida pelo sistema a partir da webcam, segmentando a região retangular contendo a face na imagem. A detecção é feita pela aplicação das características Haar, contudo este processo ocorre sem a interação de um usuário. A face segmentada é encaminhada para a função de cálculo das eigenfaces, sendo esta a próxima função no fluxo para a atribuição da presença, um exemplo da marcação da ROI na interface é mostrada na Figura 3.2, a marcação é feita na imagem que é apresentada para os usuários do sistema, contudo a imagem enviada para o sistema contém somente a região da face na imagem.



Figura 3.2: ROI da face.

3.4 Eigenfaces

A obtenção da eigenfaces é feita utilizando a biblioteca Numpy (NUMPY, 2005), sendo esta uma biblioteca de código aberto contendo os mais diversos tipos de processos de computação numérica e amplamente utilizada na comunidade científica. Para os cálculos das eigenfaces deve-se considerar um tamanho padrão para as imagens que serão processadas no sistema, sendo assim, tanto as imagens do banco de faces e as imagens adquiridas pela webcam foram redimensionadas no tamanho $N = 300$ e possuem somente um canal em escala de cinza para a minimização dos cálculos (Imagem sem modificar $I : (N \times N \times 3)$, imagem modificada $I : (N \times N \times 1)$) 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, um exemplo da imagem de treinamento pode ser vista na figura 3.4.

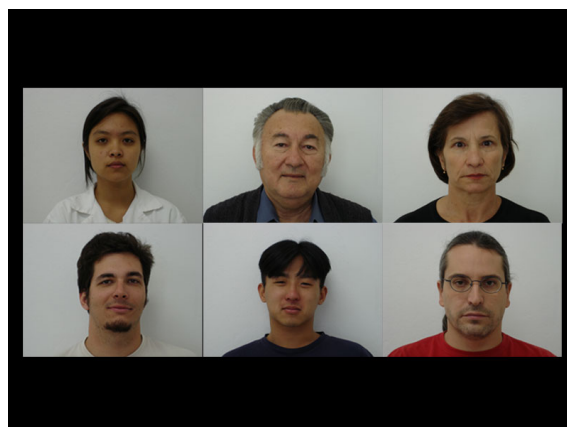


Figura 3.3: Faces sem tratamento pertencentes ao treino.

As imagens são ser normalizadas dividindo os seus valores de intensidade de cada pixel pelo valor de 255 para facilitar os cálculos. Após este processo de redimensionamento 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

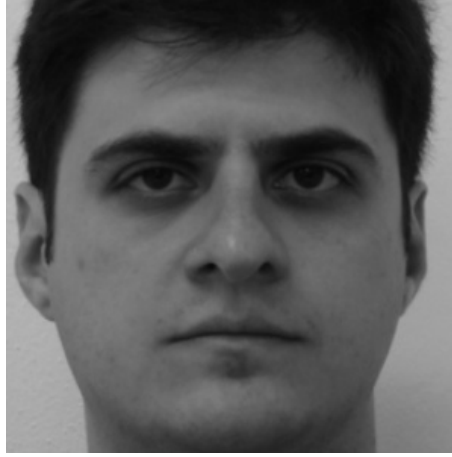


Figura 3.4: Face tratada pertencente ao treino.

uma imagem Ψ na equação 3.1.

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (3.1)$$

A imagem gerada é um modelo de face contendo as características mais comuns dentre as faces, o resultado deste processo pode ser notado na figura 3.5.



Figura 3.5: Face média Ψ .

Cada face Γ pode ser subtraída da face média Ψ para assim formar um vetor de diferenças Φ presente na equação 3.2, sendo estas as diferenças mais distintas de cada face, essa diferença pode ser notada na figura 3.6.

$$\Phi_i = \Gamma_i - \Psi \quad (3.2)$$

Os autovetores e autovalores podem ser obtidas a partir da matriz de covariância do vetor AA^T como pode ser visto na equação 3.3, sendo este um vetor de Φ , 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 ($M \times M$).



Figura 3.6: Faces Φ .

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = AA^T \quad (N^2 \times N^2) \quad (3.3)$$

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 3.4.

$$A^T A v_i = \mu_i v_i \Rightarrow AA^T A v_i = \mu_i A v_i \Rightarrow C A v_i = \mu_i A v_i \Rightarrow \text{onde } \rightarrow u_i = A v_i \quad (3.4)$$

A seguir são obtidos 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 a visualização em uma imagem. Alguns exemplos de eigenfaces obtidas podem ser vistas na figura 3.7.



Figura 3.7: Eigenfaces.

Finalmente, após a geração das eigenfaces, é necessário calcular os pesos de cada eigenface para as diferentes faces de treinamento, os pesos calculados para cada face são obtidos utilizando as eigenfaces transpostas vezes a face Φ como na equação 3.5. Durante a fase de supervisão do sistema, após os pesos Ω serem gerados, eles são inseridos no banco de dados já que durante a fase de atribuição da presença somente é necessário o cálculo dos pesos Ω da face adquirida.

$$w_j = u_j^T \Phi_i \quad (3.5)$$

O conjunto de pesos w_j é distinto para cada face, a soma dos pesos mais a face média Ψ , resulta na face original Φ , contudo essa reconstrução pode não ocorrer do jeito esperado, já que cada rosto possui uma forma própria mesmo que elas sejam alinhadas. A representação das faces no "espaço de faces" é feita pelo vetor de pesos Ω para cada face como na equação 3.6

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ w_3^i \\ \vdots \\ w_K^i \end{bmatrix}, \quad i = 1, 2, \dots, M \quad (3.6)$$

3.5 Reconhecimento

O processo de reconhecimento da face do aluno é feito por meio da classificação utilizando eigenface como representação numérica da face. A função de reconhecimento da face acontece somente no modo de atribuição de presença do aluno, já que no modo de supervisão é necessária somente a criação das eigenfaces, os cálculos e processos para classificação das eigenfaces foram realizados utilizando a biblioteca Numpy.

O aluno que acessa o sistema apresentando-se ao reconhecimento, já teve previamente seus dados caracterizados e inseridos no banco de dados do sistema. A imagem da face de entrada no reconhecimento não possui os seus pesos Ω calculados em relação as eigenfaces do sistema, assim se faz necessário o cálculo dos pesos Ω para fazer o reconhecimento da face de entrada com algumas das faces cadastradas no sistema. Este processo é uma representação da face nova, de fora do conjunto de treinamento, no "espaço de faces" normalizando ela com a face média Ψ como na equação 3.7 (a nova face já deve estar no formato de Γ para ser representada).

$$\hat{\Phi} = \Gamma - \Psi \quad (3.7)$$

Para a projeção das eigenfaces é necessário calcular o vetor de pesos Ω desta face Γ , utilizando as eigenfaces obtidas anteriormente, essa representação pode ser vista na equação 3.8

$$w_i = u_i^T \hat{\Phi} \quad \Omega = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_K \end{bmatrix} \quad (3.8)$$

O vetor Ω da nova face, é o menor erro euclidiano entre os pesos da nova face menos os pesos de cada face do conjunto de treinamento, sendo este conjunto as eigenfaces obtidas pelo supervisor do sistema após o cadastro dos usuários. Este erro deve ser o menor possível e estar abaixo de um valor de limiar para qual a face de entrada seja considerada como pertencente a um indivíduo cadastrado no sistema com seus dados já inseridos no banco de dados. A equação 3.9 representa essa distância entre a face de entrada e cada face do conjunto de treino.

$$e_\Omega = \min_l \|\Omega - \Omega^l\| \quad (3.9)$$

O valor de limiar pode ser escolhido arbitrariamente ou perante um critério de cada aplicação, segundo (SLAVKOVIC; JEVTIC, 2012), não há uma fórmula para o cálculo de um valor de limiar, o que se pode fazer é calcular o erro mínimo da imagem de teste com o treinamento, obtido a partir da Eq. 3.9, e calcular utilizando a equação 3.10.

$$\theta = 0,8 * \max(rast) \quad (3.10)$$

As seguintes considerações foram feitas utilizando os valores de limiar a partir de experimentação:

- Se o menor valor de $e_{\Omega} < \theta_{\Omega}$ a face pode pertencer ao sistema;
- Se o menor valor de $e_{\Phi} < \theta_{\Phi}$ a face é pertencente ao sistema;
- Se o valor de e_{Ω} considera uma face diferente de e_{Φ} , o valor de e_{Φ} é levado em consideração;
- Se o menor valor de $e_{\Omega} \geq \theta_{\Omega}$ a face não pertencer ao sistema;

A imagem da face reconhecida como pertencente a um indivíduo cadastrado no banco de dados do sistema é apresentada na janela da interface. Desta forma o aluno pode conferir se foi reconhecido ou não. Caso a face apresentada na interface não seja do indivíduo ele poderá entrar com o seu usuário na janela que foi apresentado a face reconhecida, gerando assim um aviso para o supervisor.

3.6 Banco de Dados

O sistema proposto deve armazenar e recuperar as imagens e valores utilizados pelo sistema. As primeiras versões do sistema utilizaram as imagens armazenadas localmente na máquina de desenvolvimento do projeto. Contudo este tipo de solução se torna inadequado, pois pode esgotar o armazenamento interno do RaspberryPi, sendo assim o armazenamento das informações foi dividido em duas formas: o armazenamento das informações dos usuários e o armazenamento das imagens.

As imagens utilizadas para o treinamento no presente trabalho foram retiradas do banco de imagens FEI face database, ver (FEI, 2012), sendo este um banco de imagens desenvolvido pelo Laboratório de Processamento de Imagens do Departamento de Engenharia Elétrica do Centro Universitário da FEI. Esse banco contém 14 imagens diferentes de cada um dos 200 indivíduos diferentes, obtendo um total de 2800 imagens. Dentre os 200 indivíduos pertencentes ao banco, para o treinamento do presente trabalho somente 49 indivíduos foram utilizados, sendo 1/4 do valor total. As imagens do banco de faces possuem uma diversidade de indivíduos, tendo diferenças de idade, diferenças de sexo, diferenças de etnias, assim sendo possível gerar uma diversidade de PCAs.

Segundo (TURK; PENTLAND, 1991), o fundo da cena na imagem onde a face deve ser caracterizada por eigenface pode acarretar problemas significantes, já que as eigenfaces não fazem distinção do que é ou não uma face. Sendo assim somente as imagens de face frontal foram utilizadas para o formar um conjunto de treinamento e teste. Todavia é necessário segmentar somente o rosto de cada indivíduo.

As imagens que são utilizadas pelo sistema projetado devem ser alocadas em um local que seja de fácil acesso e armazenamento, então foi pensado em utilizar um banco que tivesse uma liberdade de inserir as informações das imagens sem a necessidade de um banco complexo. Com isso o banco de dados MongoDB foi utilizado, ver (MONGODB,

2020), sendo o MongoDB 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). O MongoDB ainda pode ser utilizado com repositórios locais ou repositórios armazenados em nuvem de forma gratuita (conforme utilização do armazenamento e acesso repetidos dos dados). Dentre suas vantagens se encontram, por exemplo o fato de não ser necessário estabelecer as condições relacionais para começar o armazenamento de dados, obtendo melhor desempenho em relação as suas consultas, já que tudo esta dentro de um único documento.

Devido à quantidade de arquivos que são necessários para o funcionamento do projeto, as informações dos arquivos foram salvas no MongoDB e os arquivos propriamente ditos foram armazenados no Google Drive, sendo este um serviço de armazenamento e sincronização de arquivos desenvolvido pela empresa Google. O acesso aos bancos de dados ocorre a partir da configuração e utilização de arquivos de configuração, que contenham os usuários e senhas para acesso, como a seguir:

- MongoDB: A configuração do acesso é feita a partir da função *MongoClient*, sendo necessário passar como parâmetro a conexão do repositório remoto ou local.
- Google Drive: A configuração do acesso é feito a partir do arquivo *settings.yaml*, na qual consta as informações necessário, como por exemplo: informações da conta onde será feita a autenticação. A utilização do arquivo de conexão faz com que não seja necessário a autenticação manual sempre que algum processo fosse executado.

O acesso ao MongoDB para o armazenamento das informações, é feito a partir da biblioteca *Pymongo*, onde cada face tem as seguintes informações armazenadas: ID do Mongo, Nome da Pessoa e ID do Google Drive.

- O *ID do Mongo* é um identificador gerado automaticamente, sendo uma chave primária igual a de um banco de dados relacional, onde este é um identificador único.
- O *ID do Google Drive* é o identificador único gerado automaticamente quando o arquivo com as informações da imagem da face são salvos.

Para o armazenamento dos arquivos no Google Drive, o seu acesso é feito a partir da biblioteca *Pydrive*, onde cada arquivo tem as seguintes informações armazenadas: Valor de Φ da face, valor de Ψ da face e valor das eigenfaces;

O sistema acessa o banco de dados em ambos os modos de operação do sistema:

- No modo de supervisão os bancos são acessados para criar e alterar os dados armazenados.
- No modo de atribuição da presença os bancos são acessados para a consulta das informações das eigenfaces e inserção da presença.

Os valores são armazenados em um arquivo *JSON*, sendo o JSON uma notação de objetos JavaScript, onde é um formato de texto independente de linguagem de programação, pode ser interpretado pelas mais diversas linguagens. Para o armazenamento dos valores poderiam ser utilizadas outras formas de salvar o arquivo, contudo foi escolhido o arquivo JSON para a facilitação ao acesso das informações, já que as mesmas são armazenadas em formato de lista, sendo mais fácil de fazer a transformação para vetores da biblioteca *Numpy*.

3.7 Resultado do reconhecimento

Assim que uma face passa pelo processo de reconhecimento, a imagem da face reconhecida é adquirida no banco de dados e apresentada na interface do sistema. É fornecido pelo sistema o resultado da operação, onde a finalização é o atestado de presença de um aluno. Este processo ocorre sempre que a face do indivíduo já foi adquirida e tratada pelo sistema, contudo existem duas possibilidades diferentes nos resultados obtidos, que são tratadas de formas diferentes, sendo elas:

- Inserção da presença no sistema

Após a aquisição da imagem, segmentação da face do aluno, cálculo de seus pesos Ω e classificação de sua face, a presença é atribuída ao aluno se caso ele seja pertencente ao banco de faces do sistema. Assim que essa operação tem a sua finalização realizada, o sistema retorna procurando um novo indivíduo para que o processo possa ser repetido enquanto a aula/evento não seja finalizada.

- Geração do aviso para o supervisor

A geração do aviso para o supervisor ocorre caso o aluno não seja reconhecido pelo sistema, sendo assim se faz necessário que o aluno coloque manualmente o seu nome no sistema, com isso sua presença pode ser computada e o aviso é gerado para o supervisor do sistema, com as informações inseridas pelo aluno, juntamente com a foto adquirida no momento em que a presença foi realizada pelo sistema. Com esse aviso gerado pelo sistema, o supervisor pode realizar os procedimentos necessários para a inclusão do indivíduo no sistema.

3.8 Integração do Sistema

O sistema com todas as funções integradas pode ser utilizado em qualquer equipamento que possua acesso à internet, seja instalada a linguagem Python, para que o sistema seja executado e manutenção, e possa ter conectada uma webcam (ou qualquer tipo de obtenção de imagem previamente configurado para realizar a função de uma webcam, como por exemplo: celular, câmera fotográfica e webcam embutida). Há necessidade de realizar as configurações na máquina para que o banco de dados possa ser acessado corretamente. Dentro do sistema proposto no presente trabalho, a utilização de uma Raspberry Pi foi pensada devido a sua portabilidade em relação a tamanho e peso, além de necessitar pouca configuração prévia para a sua utilização. Com isso o sistema possui o seu diagrama de funcionamento representado na Figura 3.8.

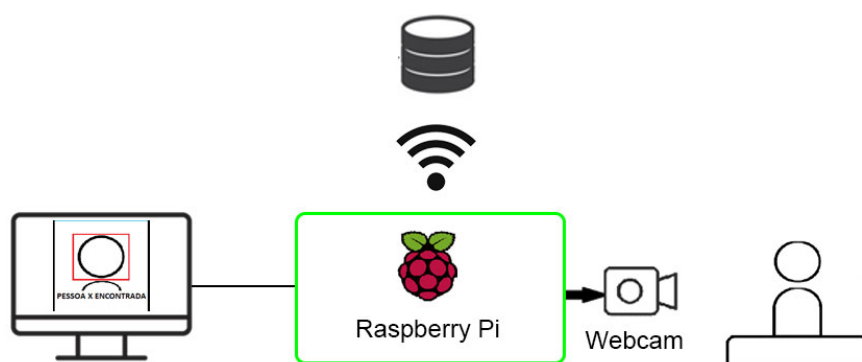


Figura 3.8: Diagrama do sistema.

4 RESULTADOS E EXPERIMENTOS

Neste capítulo são discutidos os resultados obtidos a partir dos testes aplicados com o reconhecimento utilizando as eigenfaces e o desempenho do sistema em função da configuração dos bancos de dados externos.

4.1 Descrição dos Testes com as Eigenfaces

Os testes realizados com as eigenfaces obtidas são divididos em quatro testes distintos, sendo eles: o percentual de acerto no reconhecimento, a determinação da quantidade necessária de eigenfaces para o reconhecimento, o efeito do desvio padrão nos pesos de cada face e a comparação do desempenho entre os parâmetros de reconhecimento.

4.1.1 Percentual de Acerto no Reconhecimento

O sistema proposto utiliza o reconhecimento facial para atribuir a presença para o aluno, todavia é necessário que o aluno tenha a imagem da sua face adquirida no modo de atribuição de presença reconhecida como a face cadastrada no banco de dados. Assim o teste realizado tem como objetivo obter a porcentagem de acerto do reconhecimento utilizando o valor de erro mínimo da Eq. 3.9.

O teste de reconhecimento foi realizado utilizando 50 indivíduos diferentes, onde a imagem da face do indivíduo é espelhada gerando duas imagens diferentes para cada indivíduo e totalizando 100 imagens para o teste de reconhecimento. Contudo cada imagem do teste possui distúrbios, sendo estes distúrbios as modificações faciais no rosto de cada pessoa, como por exemplo: o sorriso, os olhos mais fechados e as expressões sérias ou felizes.

A aferição do resultado do teste foi feita visualmente/manualmente, onde a imagem de entrada do teste e a imagem do banco de dados indicada pelo sistema como pertencente ao indivíduo são apresentadas em uma janela, como na Figura 4.1. Assim se uma face de teste fosse reconhecida visualmente como pertencente ao mesmo indivíduo indicado pelo sistema, ou seja, correspondente a imagem da face armazenada no banco de dados o status de acerto era atribuído ao teste, caso contrário, o status de erro era atribuído manualmente.

A primeira face da Figura 4.1(a) é a face de entrada que foi utilizada como teste de reconhecimento e a segunda face é a face que foi reconhecida pelo sistema como a que possui o menor valor de erro, sendo a segunda a imagem correspondente ao indivíduo no banco de dados. Entretanto a Figura 4.1(b) mostra uma face de teste que foi reconhecida erroneamente com outra face do banco de dados. No primeiro teste realizado, o valor de K era igual a 50, sendo K a quantidade de eigenfaces utilizadas para realizar o reconhecimento, assim com este valor foi obtido um total de 80% de faces reconhecidas.



Figura 4.1: Face de teste, Face reconhecida do banco de dados: (a) Indivíduo reconhecido, (b) Indivíduo não reconhecido

4.1.2 Determinação do valor de K

O valor do parâmetro K que é o número de melhores autovetores com os maiores autovalores para serem utilizados no sistema, não possui uma faixa de valor indicado na literatura. Sendo assim é necessário verificar o valor ótimo para a utilização no sistema proposto.

O teste para a obtenção do percentual de reconhecimento foi realizado com um valor de K igual a 50, contudo segundo (TURK; PENTLAND, 1991), utilizar as K melhores eigenfaces esta relacionado a eficiência computacional e eficiência do reconhecimento, com isso pensou-se em verificar a relação entre K e o desempenho do sistema. Desta forma otimizar o sistema e aumentando a taxa de acerto. Assim foi realizada um teste para determinar o melhor valor de K para o reconhecimento.

O teste para a determinação do valor ótimo de K foi feito como descrito no item 4.1.1. Foram feitas 10 baterias de teste, variando o valor de K entre 10 e 100 com passo de valor 10 e verificando a taxa de acerto do sistema. Os percentuais de acerto de cada valor de K são apresentados na Figura 4.2 e os percentuais de erro na Figura 4.3.

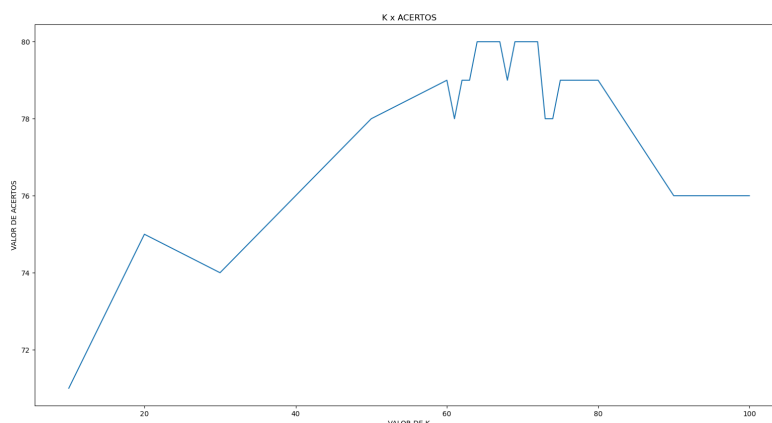


Figura 4.2: Gráfico K x Acertos.

A curva do gráfico da Figura 4.2 mostra que o maior número de acerto foi obtido com K na faixa entre 60 e 80. Assim foi escolhido o valor de $K = 70$, sendo utilizadas as melhores 70 eigenfaces obtidas previamente no processo de reconhecimento.

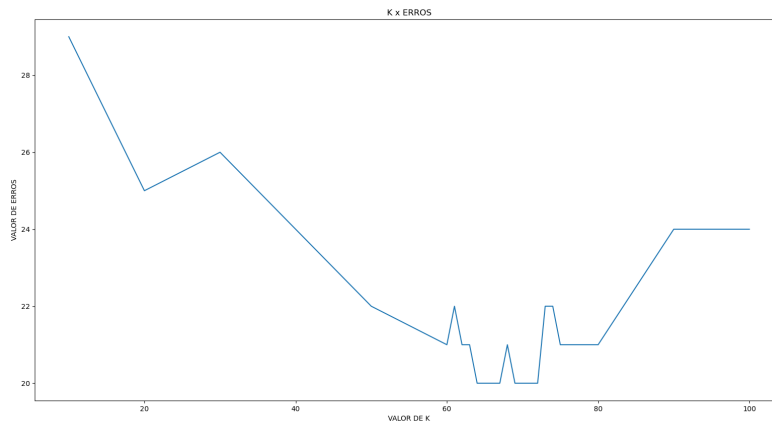


Figura 4.3: Gráfico K x Erros.

4.1.3 Efeito do desvio padrão nos pesos Ω

O desvio padrão dos valores dos pesos Ω s da imagem de faces semelhantes tem valores aproximados quando projetados no espaço de faces, portanto pode ser usado para encontrar faces que possuem características semelhantes. A relação entre desvio padrão e o índice dos pesos pode ser notado na Figura 4.4. Esta relação pode ser notada também no desvio padrão das imagens utilizadas no conjunto de treinamento, onde os pontos relativos a faces que possuem características semelhantes se aproximam no gráfico resultando em um agrupamento no gráfico da Figura 4.5.

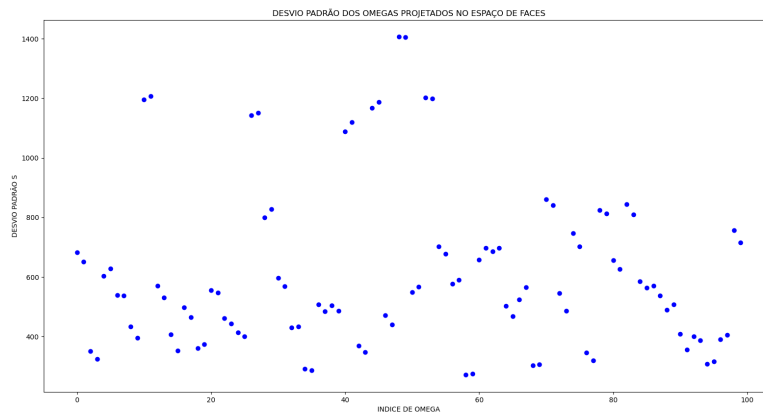


Figura 4.4: Desvio Padrão dos Omegas projetados no espaço de faces.

4.1.4 Comparação do desempenho entre os parâmetros de reconhecimento (Erro mínimo de Ω x Erro mínimo de Φ)

A taxa de acerto de reconhecimento da face dos indivíduos obtida utilizando o parâmetro Erro mínimo de Ω da Eq. 3.9 foi de 80%. Contudo foi proposto a utilização do parâmetro Erro mínimo de Φ da Eq. 4.1, para investigar a sua utilização como parâmetro de reconhecimento e a possibilidade de aumentar a taxa de acerto do sistema. Aqui são apresentados os testes e resultados da aplicação destes dois parâmetros e a comparação

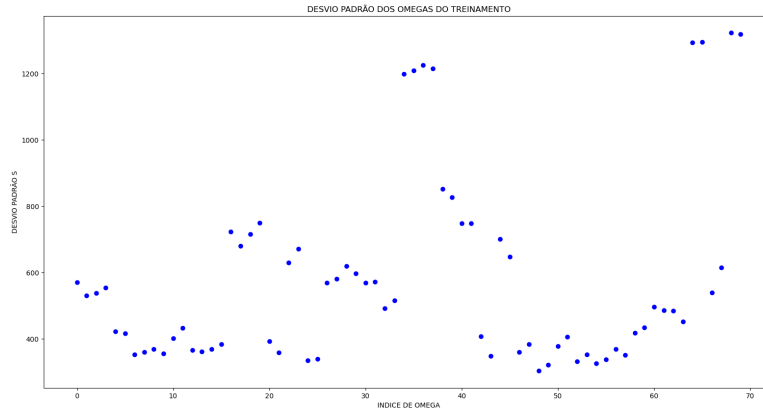


Figura 4.5: Desvio Padrão dos Omegas no espaço de faces.

do desempenho de cada um destes parâmetros.

Os testes utilizando o Erro mínimo de Φ foram feitos da mesma forma descrita em 4.1.1. A utilização do parâmetro Erro mínimo de Φ , Eq. 4.1 resultou em um aumento de 10% de acertos em comparação com os primeiros testes de reconhecimento utilizando o parâmetro de Erro mínimo Ω .

$$e_{\Phi} = \min \|\hat{\Phi} - \Phi_i\| \quad (4.1)$$

Os gráficos das Figuras 4.6 a 4.11 são os resultados obtidos na realização do teste de reconhecimento, contudo nestes gráficos também à a representação dos testes realizados a partir do reconhecimento automático. As legendas das figuras são descritas em pares, onde os círculos azuis e os losangos vermelhos representam o teste realizado sem a visualização manual e os círculos verdes e losangos amarelos representam o teste realizado igualmente no item 4.1.1.

O teste de reconhecimento automático foi realizado a partir da utilização de um limiar obtido na observação do gráfico da Figura 4.6, na qual mais de 10% das imagens das faces não reconhecidas possuíam um valor de Erro mínimo de Φ maior do que 30. Os resultados dos gráficos as Figuras 4.7 a 4.11 são a variação do limiar do Erro mínimo Φ com passo de valor 1.

Os gráficos das Figuras 4.6 a 4.11 possuem diferenças de erro entre a o reconhecimento manual e o reconhecimento automático, onde a imagem de uma face é atribuída como reconhecida devido ao seu valor de Erro mínimo de Φ estar abaixo do limiar escolhido, causando assim o efeito de falso e verdadeiro negativo.

Os valores da faixa de limiares utilizados para o Erro mínimo de Φ foi verificado realizando os mesmos testes descritos anteriormente, contudo foi utilizado imagens de faces que não pertenciam ao conjunto de treinamento armazenado no banco de dados. Assim os resultados dos gráficos das Figuras 4.12 a 4.17 foram obtidos.

Os gráficos das Figuras 4.12 a 4.17 possuem a mesma estrutura de cor e forma do teste anterior, variando o valor do limiar de Erro mínimo de Φ com o mesmo passo de valor 1. Com os resultados do teste foi possível notar que os valores de faces que não pertencem ao conjunto de treinamento possuem um valor de Erro mínimo de Φ superior a faixa de 30 a 35. Todavia algumas imagens das faces possuem um valor dentro da faixa de limiar escolhido.

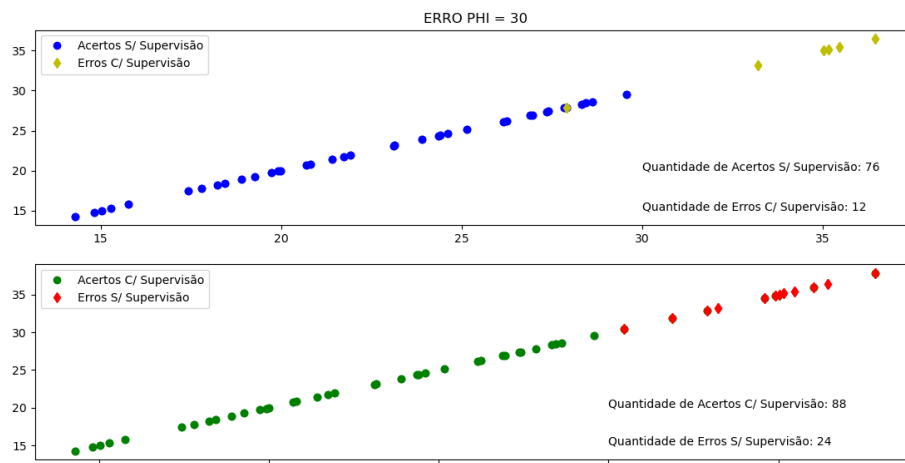


Figura 4.6: Faces reconhecidas do treinamento utilizando Erro $\Phi = 30$.

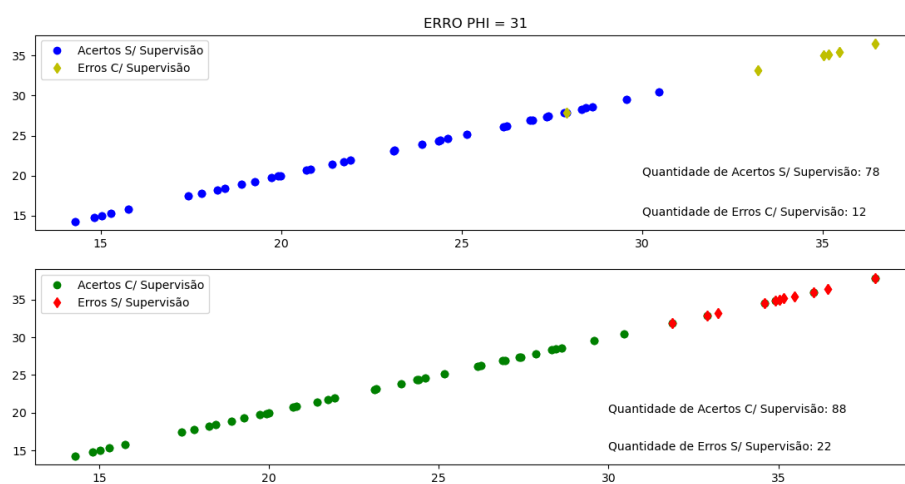


Figura 4.7: Faces reconhecidas do treinamento utilizando Erro $\Phi = 31$.

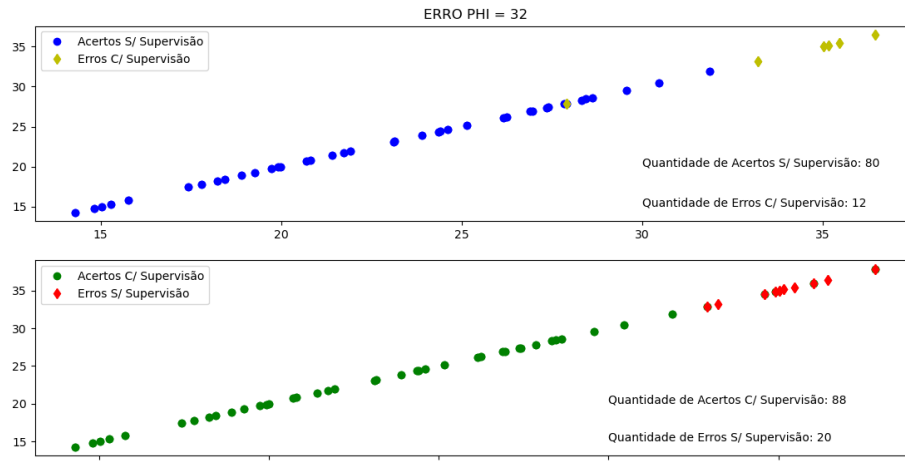


Figura 4.8: Faces reconhecidas do treinamento utilizando Erro $\Phi = 32$.

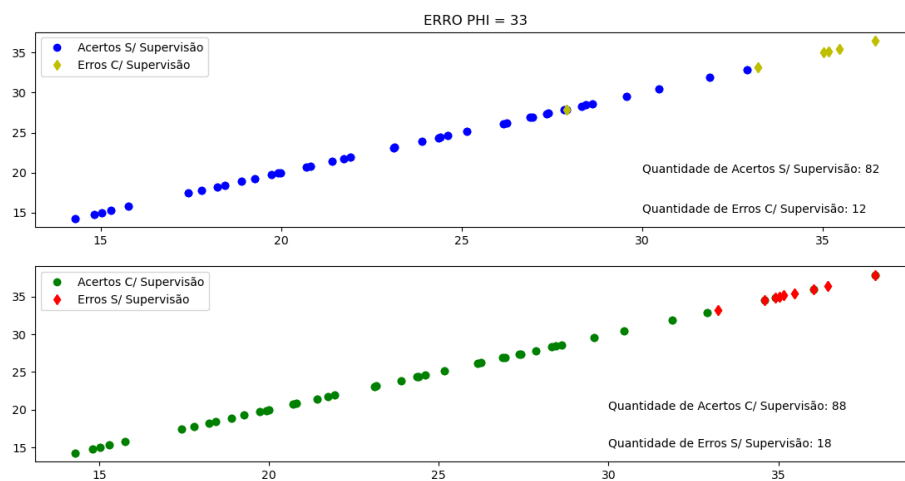


Figura 4.9: Faces reconhecidas do treinamento utilizando Erro $\Phi = 33$.

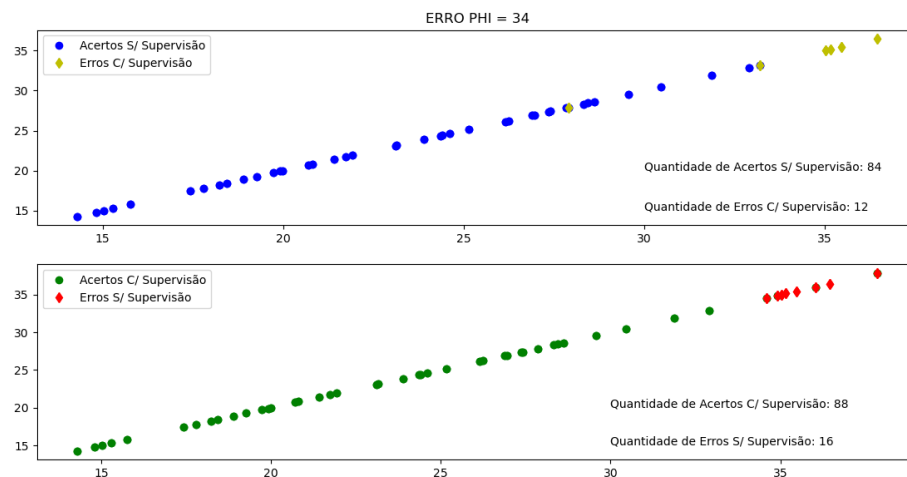


Figura 4.10: Faces reconhecidas do treinamento utilizando Erro $\Phi = 34$.

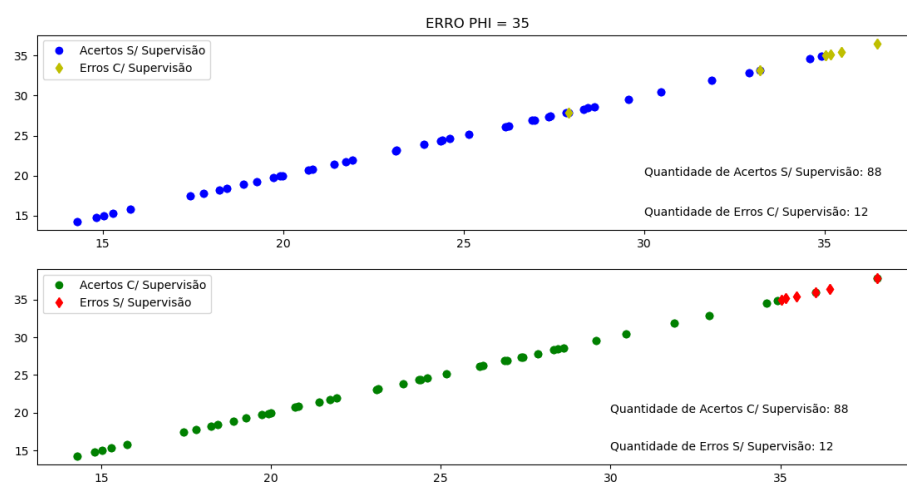


Figura 4.11: Faces reconhecidas do treinamento utilizando Erro $\Phi = 35$.

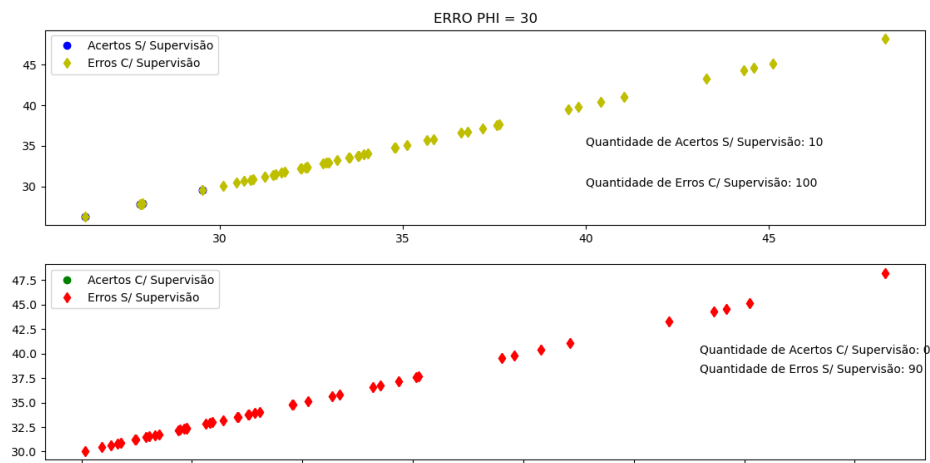


Figura 4.12: Fases reconhecidas de fora do treinamento utilizando Erro $\Phi = 30$.

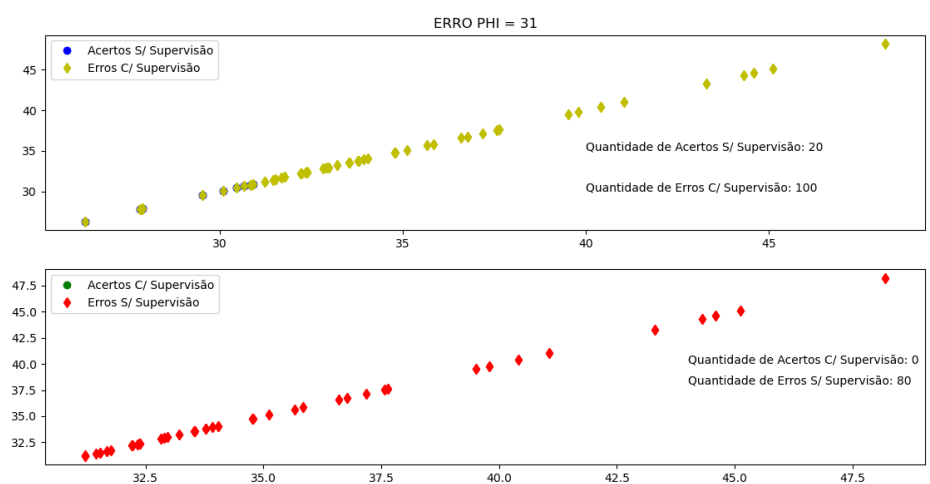


Figura 4.13: Fases reconhecidas de fora do treinamento utilizando Erro $\Phi = 31$.

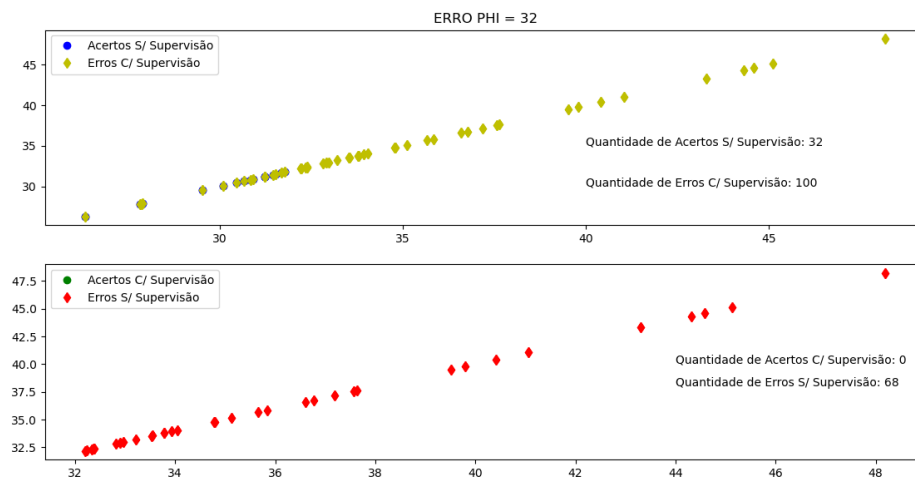


Figura 4.14: FACES reconhecidas de fora do treinamento utilizando Erro $\Phi = 32$.

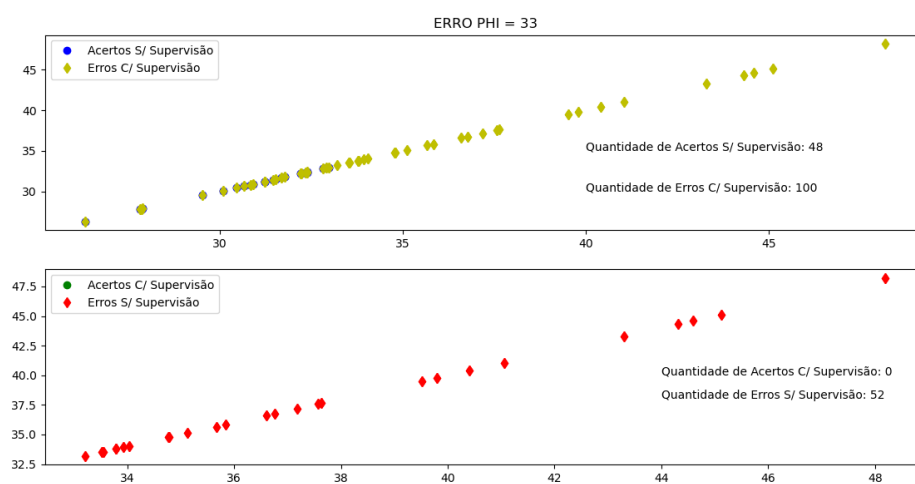


Figura 4.15: FACES reconhecidas de fora do treinamento utilizando Erro $\Phi = 33$.

As faces acabam sendo consideradas como pertencentes ao conjunto de treinamento devido a igualdade de certas características humanas, sendo este um fator que deve ser levado em consideração para fazer uma análise de grupos de faces.

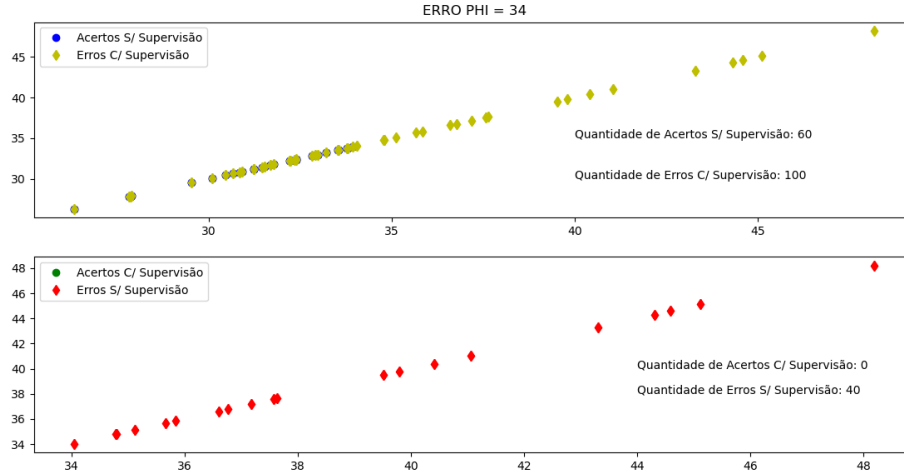


Figura 4.16: Faces reconhecidas de fora do treinamento utilizando Erro $\Phi = 34$.

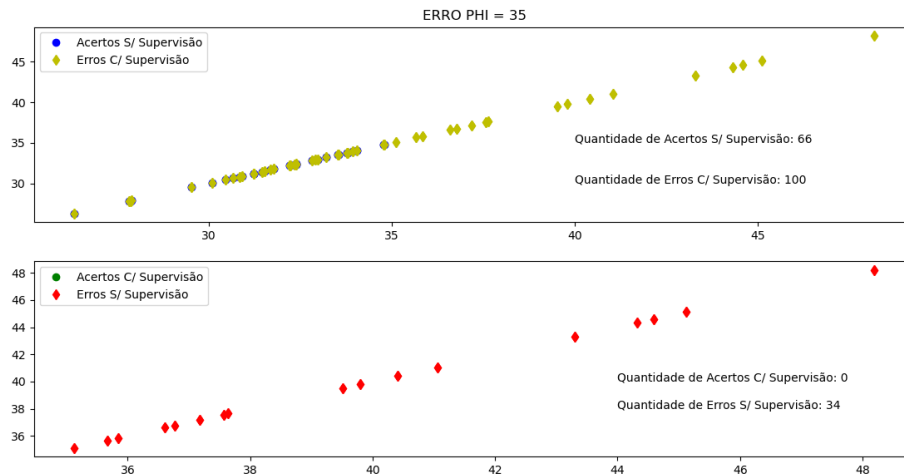


Figura 4.17: Faces reconhecidas de fora do treinamento utilizando Erro $\Phi = 35$.

A análise dos valores de falsos e positivos negativos que podem ocorrer com o aumento do valor de *Erro Phi* são descritos na tabela 4.1, onde **FN** é a quantidade de valores de "*Falsos Negativos*" das imagens das faces que não foram reconhecidas devido ao seu valor de Erro mínimo Φ estar acima do limiar, **FP** é a quantidade de valores de "*Falsos Positivos*" das imagens das faces que foram reconhecidas devido ao seu valor de Erro mínimo Φ estar dentro do limiar, σ é a soma dos acertos e erros subtraídos das quantidades de falsos durante o teste, ver Eq.4.2, **Acertos** é a quantidade de imagens de faces reconhecidas automaticamente utilizando o valor de limiar de Erro mínimo Φ e **Erros** é a quantidade de imagens de faces não reconhecidas automaticamente utilizando o valor de limiar de Erro mínimo Φ .

$$\sigma = (Acertos - FP) + (Erros - FN) \quad (4.2)$$

FN	FP	σ	Acertos	Erros
$e_{\Phi} = 30$				
14	2	84	76	24
$e_{\Phi} = 31$				
12	2	86	78	22
$e_{\Phi} = 32$				
10	2	88	80	20
$e_{\Phi} = 33$				
8	2	90	82	18
$e_{\Phi} = 34$				
8	4	88	84	16
$e_{\Phi} = 35$				
4	4	92	88	12

Tabela 4.1: Falsos positivos e negativos

Pode-se notar que conforme o valor de e_{Φ} vai aumentando, a quantidade de falsos negativos tende a diminuir, sendo assim pode-se notar que escolher somente um valor de limiar é uma limitação, pode utilizar ao invés de somente um valor ser uma faixa, sendo assim a faixa vai de 30 a 35. Com todos os valores de limiares, face Ψ e eigenfaces, pode-se utilizar isto para a verificação no sistema.

4.2 Teste de tempo de resposta do Banco de Dados

O sistema proposto possui a necessidade de um modo de armazenar e recuperar as imagens e valores utilizados pelo sistema. O acesso das imagens na máquina de desenvolvimento do projeto não possui atraso para a obtenção de cada imagem, contudo caso exista algum tipo de perda do sistema por meios físicos, as imagens serão perdidas e será necessário reconfigurar o sistema. Assim a utilização de um meio de armazenamento externo foi adotada como uma forma mais prática de salvar as informações do sistema. Todavia existem atrasos para que seja feita a recuperação das informações que foram salvas externamente.

O processo de armazenamento é realizado com a utilização de dois bancos de dados diferentes, sendo eles: o MongoDB para o armazenamento das informações das imagens armazenadas e o Google Drive para o armazenamento das imagens. O armazenamento ocorre em três etapas, sendo elas: obter as informações da face (valores de Φ e Ω de cada face individualmente), salvar as informações obtidas no Google Drive, obter o ID do Google Drive e salvar no MongoDB juntamente com o nome de cada pessoa. Um exemplo deste processo é apresentado na Figura 4.18.

A análise de uma face de entrada no sistema necessita dos valores que foram armazenados nos bancos de dados, assim a busca dos valores de cada face no banco de imagens é realizada. Os valores de Ω , que foram armazenados previamente no banco de dados, são utilizados para a etapa de reconhecimento, a imagem da face que foi reconhecida com o valor Ω que representa a sua face é enviado para o sistema.



Figura 4.18: Armazenamento das imagens a partir dos Bancos de Dados.

Os arquivos foram divididos em três classes: (1) a face média Ψ (contém somente um arquivo), (2) as eigenfaces (contém setenta arquivos diferentes) e, (3) as imagens das faces Φ (contém cento e noventa e seis arquivos diferentes).

A aquisição dos dados de tempo de recuperação das imagens armazenadas nos bancos de dados acontece da seguinte forma:

- O identificador único armazenado no MongoDB é acessado pelo sistema.
- O tempo atual do sistema é adquirido e armazenado em uma variável.
- O identificador é utilizado para o acesso do arquivo armazenado no Google Drive.
- As informações de cada arquivo são armazenadas em uma variável.
- O novo tempo do sistema é armazenado em uma nova variável.
- O tempo gasto para a operação é calculado utilizando a subtração dos tempos obtidos.

Os resultados da recuperação das imagens armazenadas estão descritos na tabela 4.2, na qual tem a quantidade de arquivos que foram recuperados e o tempo gasto para que a operação fosse concluída.

	Ψ	Eigenfaces	Φ
Qnt Arquivos	1	70	196
Tempo Gasto	2,2s	157s	475s

Tabela 4.2: Tempo de recuperação das imagens

Os tempos de aquisições dos arquivos estão principalmente atrelados ao acesso ao Google Drive, pois o tempo de obtenção de todos os identificadores que estão armazenados no MongoDB esta entre 1 e 2 segundos. Com um total de 635 segundos para obter todos os arquivos necessários para o funcionamento do sistema, estes arquivos devem ser carregados na inicialização do sistema e ser utilizado localmente, devido a esta quantidade de tempo necessária para fazer a requisição dos arquivos, pois para cada face que irá ser reconhecida é necessário todos os arquivos armazenados, sendo assim o tempo total de acesso ao banco de dados mais o tempo para o reconhecimento é o tempo total que o usuário deve esperar pelo reconhecimento.

5 CONCLUSÃO E TRABALHOS FUTUROS

O sistema proposto no presente trabalho possui dois modos de operação, o modo de supervisão e o modo de atribuição de presença do aluno, na qual o modo de supervisão permite a interação do supervisor com as funções de configuração do sistema e o modo de atribuição de presença do aluno que realiza as funções de obtenção da imagem da face de cada aluno, que se apresenta a interface de forma voluntária, e realiza o reconhecimento para a atribuição da presença de forma automática. A utilização do sistema é feita a partir de interfaces.

Além dos modos de operação, o sistema contém a integração com banco de dados externos para o armazenamento das imagens das faces de cada aluno cadastrado no sistema. O reconhecimento das imagens das faces foi realizado utilizando dois parâmetros diferentes durante os testes do sistema, sendo estes parâmetros o Erro mínimo Ω e Erro mínimo Φ . Contudo a utilização dos valores obtidos do Erro mínimo Ω e Erro mínimo Φ para o reconhecimento dos indivíduos através das faces devem estar abaixo de um valor de limiar para que a imagem de uma face seja reconhecida automaticamente. A determinação de um limiar para o reconhecimento é uma tarefa árdua, já que até o presente momento não há na literatura uma técnica para obtê-la ou valor recomendado para ser utilizado de forma genérica em todos os sistemas.

Ainda com a utilização do modelo de eigenfaces para o reconhecimento facial, ver seção 4.1, pode-se perceber que os valores variam dependendo da quantidade de eigenfaces utilizadas, sendo assim a quantidade de eigenfaces acaba sendo um fator de otimização que deve ser trabalhado. Os valores ótimos obtidos para K no sistema foram: $K = 70$ e a faixa de valores $e_{\Phi} = 30$ a 35 .

Os testes realizados com os valores de Erro mínimo Ω e Erro mínimo Φ para o reconhecimento, mostram que com a utilização do Erro mínimo Φ resulta em uma taxa de acerto no reconhecimento das faces 10% maior do que a utilização do Erro mínimo Ω . A faixa de percentual de acerto do Erro mínimo de Φ foi de 76% a 88% de acerto do conjunto de faces utilizadas, onde as faces utilizadas possuíam modificações mínimas nas expressões faciais. Em síntese o indivíduo que quer ser reconhecido pelo sistema não irá apresentar adversidades para o seu reconhecimento acrescentando elementos estranhos à face, já que ele quer a aprovação no sistema. Portanto a taxa de acerto obtida nos testes pode ser considerada adequada à aplicação proposta.

A utilização de um banco de dados em nuvem pode ser uma solução viável para que os dados não sejam perdidos em caso de falhas do equipamento que executa o sistema. Contudo o tempo de resposta do banco de dados devem ser levados em consideração na sua utilização. Os tempos de resposta obtidos no sistema proposto teve um total de 635 segundos para a recuperação de todos os 267 arquivos armazenados. O tempo de resposta do banco de dados pode aumentar em relação a quantidade de arquivos.

O sistema proposto foi desenvolvido para a possibilidade de ser embarcado, para que seja levado para os locais necessários para a atribuição da presença do aluno. Assim a utilização de uma Raspberry Pi para embarcar o sistema proposto, torna simples o deslocamento do sistema para os locais em que se faz necessário o seu uso, já que uma das vantagens da utilização da Raspberry Pi é a possibilidade de alimentação por meio de baterias externas.

Os testes do reconhecimento facial realizados durante o desenvolvimento do sistema proposto, mostraram que as imagens das faces que eram reconhecidas erroneamente possuíam características semelhantes. Assim uma proposta para trabalhos futuros seria a realização de uma análise dos resultados obtidos das faces em grupos de características semelhantes, separando as eigenfaces de modo a representar um grupo de faces ao invés de todo o conjunto de faces do sistema. Desta forma seria adotada uma pré-classificação, antes do reconhecimento.

A interação do supervisor com o sistema deverá ser diminuída no futuro se redes neurais e inteligência artificial forem aplicadas, principalmente nos processos de aquisição e cadastro de novas imagens no sistema que faria a obtenção das imagens da face e os pesos com menor interação para a supervisão humana.

REFERÊNCIAS

FEI. **FEI Face Database**. (Acesso em 20 de Março, 2019), <https://fei.edu.br/cet/facedatabase.html>.

GONZALEZ, R. C. **Processamento Digital de Imagens**. Av. Ermano Marchetti, 1453, São Paulo, São Paulo: Pearson Prentice Hall, 2010.

MISHRA, S. P.; SARKAR, U.; TARAPHDER, S.; DATTA, S.; SWAIN, D. P.; SAIKHOM, R.; PANDA, S.; LAISHRAM, M. **Multivariate Statistical Data Analysis-Principal Component Analysis (PCA)**. 2017.

MONGODB. **MongoDB**. (Acesso em 11 de Dezembro, 2020), <https://www.mongodb.com>.

NUMPY. **About Us**. (Acesso em 02 de Janeiro, 2021), <https://numpy.org>.

OPENCV. **About OpenCV**. (Acesso em 2 de Janeiro, 2021), <https://opencv.org/about/>.

PYTHON. **Python**. (Acesso em 11 de Dezembro, 2020), <https://www.python.org>.

RASPBERRYPI. **About Us**. (Acesso em 02 de Janeiro, 2021), <https://www.raspberrypi.org>.

SECURITY. **Pandemia expande uso do reconhecimento facial para controles de acesso**. (Acesso em 02 de Janeiro, 2021), <https://revistadigitalsecurity.com.br/pandemia-expande-uso-do-reconhecimento-facial-para-controles-de-acesso/>.

SIROVICH, L.; KIRBY, M. **Low-dimensional procedure for the characterization of human faces**. 1987.

SLAVKOVIC, M.; JEVTIC, D. **Face Recognition Using Eigenfaces Approach**. 2012.

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

VIOLA, P.; JONES, M. **Rapid Object Detection using a Boosted Cascade of Simple Features**. 2001.