



COORDENADORIA DE ENGENHARIA DA COMPUTAÇÃO

**JULIO CESAR DA SILVA ESTEVES
MARCELO THOMAZ DE AQUINO JUNIOR**

**ANÁLISE E RECONHECIMENTO POR VÍDEO PARA CONTROLE DE
PRESENCAS UTILIZANDO DEEP LEARNING**

Sorocaba/SP

2018

JULIO CESAR DA SILVA ESTEVES
MARCELO THOMAZ DE AQUINO JUNIOR

**ANÁLISE E RECONHECIMENTO POR VÍDEO PARA CONTROLE DE
PRESENCAS UTILIZANDO DEEP LEARNING**

Trabalho de Conclusão de Curso apresentado à
Faculdade de Engenharia de Sorocaba como
exigência parcial para a obtenção do Diploma de
Graduação em Engenharia da Computação.

Orientador: Prof. Me. André Breda Carneiro

Sorocaba/SP
2018

FICHA CATALOGRÁFICA
ELABORADA PELA "BIBLIOTECA FACENS"

E79a

Esteves, Julio Cesar da Silva.

Análise e reconhecimento por vídeo para controle de presenças utilizando deep learning / por Julio Cesar da Silva Esteves, Marcelo Thomaz de Aquino Junior.— Sorocaba, SP: [s.n.], 2018.
54f.; 29cm.

Trabalho de Conclusão de Curso (Graduação) – Faculdade de Engenharia de Sorocaba, Coordenadoria de Engenharia de Computação – Curso de Engenharia de Computação, 2018.

Orientador(a): Prof.(a) Me. André Breda Carneiro

1. Deep Learning . 2. Reconhecimento Facial. 3. D-Lib. I.Thomaz de Aquino Junior, Marcelo. II.Faculdade de Engenharia de Sorocaba III. Título.

CDD 621.39

ANÁLISE E RECONHECIMENTO POR VÍDEO PARA CONTROLE DE PRESENCAS UTILIZANDO DEEP LEARNING

Trabalho de Conclusão de Curso apresentado à
Faculdade de Engenharia de Sorocaba como
exigência parcial para a obtenção do Diploma de
Graduação em Engenharia da Computação.

Comissão examinadora:

Prof. Me. André Breda Carneiro

Prof.

Prof.

Coordenador(a):

Prof. Dra. Andrea Lúcia Braga Vieira Rodrigues

Ass. _____

Prof. Dra. Andréa Lucia Braga Vieira Rodrigues

Sorocaba/SP

2018

Dedico este trabalho primeiramente à Deus em que sempre me deu conhecimento e apoio, seguido de meus pais em que sempre colaboraram e acreditaram no meu potencial, e também à minha namorada que me deu forças para sempre continuar batalhando e também dedico ao meu colega Marcelo Thomaz, pelo comprometimento e auxílio durante essa jornada de TCC.

Dedico este trabalho a Deus, a minha família que sempre esteve ao meu lado, a minha namorada que sempre me incentivou e apoiou e ao meu amigo Julio por toda ajuda e comprometimento.

AGRADECIMENTOS

Primeiramente agradeço a Deus por me dar essa oportunidade e capacidade de executar as tarefas durante o curso tais como, provas, trabalhos, apresentações, etc.

Aos meus pais por sempre me motivar e acompanhar meus desempenhos, incentivando à conquistar os meus objetivos, e que batalharam arduamente para me dar um ensino de qualidade e à minha irmã que sempre está ao meu lado contribuindo para o meu aprendizado.

A minha dupla e os meus amigos que sempre me auxiliaram nos projetos e atividades realizadas e mantendo o foco para que os objetivos fossem alcançados.

Aos professores Tiago Pasqualini e Andre Breda Carneiro, como os dois principais colaboradores do projeto sendo diretamente ou indiretamente a contribuição mútua.

AGRADECIMENTOS

Meus agradecimentos vão primeiramente a Deus, por ter me dado forças e ter me capacitado para chegar até aqui. Agradeço também a todos familiares, amigos e colegas que de alguma forma contribuíram para que eu pudesse chegar aqui.

Agradeço a minha namorada por toda ajuda, paciência e incentivo que me deu durante toda caminhada até aqui.

E por último, mas não menos importante, aos professores Tiago Pasqualini e André Breda Carneiro por toda ajuda e apoio nessa dura etapa que é o TCC.

“Não é o mais forte que sobrevive,
nem o mais inteligente. Quem
sobrevive é o mais disposto à
mudança.”

Charles Darwin

“Não é o conhecimento, mas o ato de
aprender, não a posse, mas o ato de
chegar lá, que concede a maior
satisfação”

Carl Friedrich Gauss

RESUMO

Controle de presenças manual é um processo muito utilizado nos dias de hoje que pode ser otimizado por meio de soluções com reconhecimento de faces e, consequentemente aumentar a produtividade das aulas. O presente trabalho poderá ser aplicado a todos os cursos e disciplinas, pois tem como foco a melhoria de um processo que hoje normalmente é feito de forma não automatizada. Logo, pode-se concluir que o projeto não só contribuirá na gestão do tempo em sala de aula, como também aperfeiçoará um processo que hoje é suscetível a falhas através da aplicação de tecnologias de reconhecimento facial. Para esse projeto, será utilizado redes neurais residuais, sendo assim, pode-se obter uma precisão maior na hora das detecções conforme seu treinamento e com isso, uma boa taxa de acerto no momento de identificar se uma determinada pessoa está presente no local.

Palavras chaves: Aprendizado Profundo, Reconhecimento Facial, D-Lib.

ABSTRACT

Presence control manual is a very used process actually that can be optimized through solutions with face recognition and consequently increase the productivity of classes. This work can be applied to all courses and disciplines because it focuses on the improvement of a process that today is usually done in a non-automated way. Therefore, it can be concluded that the project will not only contribute to the management of time in the classroom, but will also improve a process that today is susceptible to failures through the application of facial recognition technologies. For this project, residual neural networks will be used, thus, one can obtain a greater accuracy at the time of the detections according to their training and with that, a good response in the moment of identifying if a certain person is present.

Key words: Deep Learning, Face Recognition, D-Lib.

LISTA DE FIGURAS

Figura 2.1 – Marcação de características da face humana	19
Figura 2.2 – Problemas que podem afetar o reconhecimento facial	21
Figura 2.3 – Exemplo de imagens com desfoque e ruídos	21
Figura 2.4 – Exemplo de rosto com barba e bigode	22
Figura 2.5 – Quatro configurações possíveis de um feature	24
Figura 2.6 – Representação Cascata de Classificadores	25
Figura 2.7 – Representação da operação LBP	26
Figura 2.8 – Operador LBP estendido	26
Figura 3.1 – Imagens com tamanhos iguais e resoluções diferentes	29
Figura 3.2 – Estrutura básica do <i>OpenCV</i>	32
Figura 4.1 – Modelo de um neurônio artificial	34
Figura 4.2 – Curva e gradiente do erro para problemas lineares	36
Figura 4.3 – Representação algoritmo <i>backpropagation</i>	36
Figura 4.4 – Modelo de camadas RNA's	38
Figura 4.5 – Estrutura de uma <i>Multilayer Perceptron</i>	39
Figura 5.1 – Etapas do sistema reconhecedor de faces	42
Figura 5.2 – Captura do <i>frame</i> durante a execução do sistema	43
Figura 5.3 – Rotina aplicando redução de tamanho da imagem (<i>resize</i>)	44
Figura 5.4 – Características de uma face utilizando Dlib	45
Figura 5.5 – Matriz de valores dos pontos identificados no rosto	45
Figura 5.6 – Reconhecimento com mais de uma pessoa	46
Figura 5.7 – Aluno reconhecido pelo sistema	47
Figura 5.8 – Arquivo texto com os alunos reconhecidos	47
Figura 5.9 – Gráfico da taxa de acerto x alunos	50

LISTA DE ABREVIATURAS E SIGLAS

BSD	<i>Berkeley Software Distribution</i>
CNN	<i>Convolutional Neural Network</i>
LBP	<i>Local Binary Patterns</i>
OpenCV	<i>Open Source Computer Vision Library</i>
DNN	<i>Deep Neural Network</i>
RGB	<i>Red Green Blue</i>
GUI	<i>Graphic User Interface</i>
LDB	Lei de Diretrizes e Bases
RA	Registro do Aluno

SUMÁRIO

1 INTRODUÇÃO	15
2 RECONHECIMENTO FACIAL	17
2.1 Conceitos Básicos	17
2.1.2 O que é reconhecimento facial	18
2.1.3 Utilização	19
2.2 Análise e Reconhecimento de Faces.....	20
2.2.1 Características	22
2.2.2 Reconhecimento e identificação.....	23
2.2.3 Algoritmos de reconhecimento	23
3 PROCESSAMENTO DE IMAGENS.....	27
3.1 O que é Processamento de Imagens	27
3.2 Análise e Captura de Imagem	28
3.3 Técnicas de Processamento de Imagens.....	29
3.4 Opencv	31
4 REDES NEURAIIS.....	33
4.1 Conceitos Básicos	33
4.1.1 O que são redes neurais	33
4.1.2 Treinamento das redes.....	35
4.2 Tipos de Redes Neurais.....	37
4.2.1 Neurônio artificial.....	37
4.2.2 <i>Perceptron</i>	37
4.2.3 Redes neurais multicamadas <i>perceptron</i> (MLP)	38
4.3 D-lib	39
5 DESENVOLVIMENTO	41
5.1 Arquitetura da Aplicação	41
5.2 Treinamento da Rede Neural.....	43
5.2.1 Análise e identificação das imagens conhecidas	43
5.2.2 Treinamento do classificador neural.....	46
5.3 Resultados	48
5.3.1 Validação.....	48

5.3.2 Comparação com o dataset	48
5.3.3 Taxa de acerto do sistema	50
6 CONCLUSÃO	51
REFERÊNCIAS.....	52

1 INTRODUÇÃO

O processamento de imagens ganha novo papel na sociedade com a popularização de equipamentos capazes de gerar, armazenar e transmitir vídeos utilizando métodos para obter informações automaticamente de vídeos e de sequência de imagens está se tornando cada vez mais comum (SOUZA, 2010).

Historicamente, em salas de aula de todo o mundo, os professores têm a obrigação de registrar a presença de alunos em suas aulas. Para atender a essa necessidade, diversos métodos foram desenvolvidos, sendo o mais comum a chamada oral da lista de presença (MATTOS, 2017).

Segundo o Projeto de Lei 4831/09 do Senado, exige do estudante universitário o comparecimento a 75% das horas letivas totais para a aprovação em cada disciplina. Esse percentual mínimo foi incluído na Lei de Diretrizes e Bases da Educação Nacional (LDB – 9.394/96) (SIQUEIRA, 2012).

Apesar de esses métodos serem eficazes, não se podem reduzir ao máximo as falhas, pois, às vezes o professor acaba pulando o nome, ou até mesmo, um aluno não ouvir a sua chamada e acabar ficando com falta irregularmente. Se utilizado cartões magnéticos, o aluno também pode cair no esquecimento e não registrar a sua presença se direcionando ao dispositivo integrado na sala de aula.

Como dito por Guilherme Mattos, 2017, a redução da intervenção humana nesse processo pode diminuir a ocorrência de erros. Assim, a detecção da presença do aluno sem que ele ou professor executem alguma ação voluntária (sem que precisassem agir conscientemente) talvez possa ser uma forma mais eficiente de registrar presença e a aplicação do reconhecimento facial durante esses processos torna o controle de presenças menos suscetível a falhas.

Esse presente trabalho visa a redução dos tempos de aulas para fazer a chamada manual, e possivelmente ser dedicado para conteúdos mais relevantes, assim aumentando a produtividade das aulas e diminuir as falhas que ocorrem durante um processo manual.

A aquisição das imagens foram obtidas através de uma câmera integrada de um notebook, desenvolvido em um sistema web onde é pego o *frame* dessa câmera e enviado para a biblioteca Dlib, que é feito o reconhecimento dos rostos e enviado

para o classificador neural onde é feito a comparação entre as faces do *frame* e as imagens que existem previamente no banco de dados.

Para a utilização da biblioteca e o desenvolvimento do classificador neural foi utilizada a linguagem de programação *Python* e posteriormente será treinado com as imagens já adicionadas previamente no banco de imagens.

O segundo capítulo teórico apresenta os conceitos básicos de reconhecimento facial, tal como a sua história, utilização em aplicações reais, além da demonstração de como é feito um processo de análise e reconhecimento de faces, bem como a sua utilização.

No terceiro capítulo é apresentado os conceitos de processamento de imagens, suas etapas e as bibliotecas que se pode utilizar para fazer os devidos tratamentos para um bom reconhecimento facial.

O quarto capítulo teórico traz informações sobre redes neurais, história de seu surgimento, algoritmos para sua utilização e como é incorporada num sistema de reconhecimentos.

O quinto e sexto capítulo expõe como foi desenvolvido o projeto, bem como os materiais utilizados, além de como foi aplicado as tecnologias estudadas e apresentadas durante o decorrer do trabalho, bem como os seus resultados apresentados nas validações e baterias de testes. Por último é mostrado a conclusão do trabalho realizado e os objetivos atingidos durante a execução do mesmo.

2 RECONHECIMENTO FACIAL

Como dito por Osvaldo de Almeida (2006), a partir da face humana, é possível extrair características como as posições dos olhos, nariz e boca, que permitem localizá-la e, além disso, possibilitam identificar uma pessoa. A fase de reconhecimento facial num sistema é extremamente importante pelo fato de toda a base de treinamento ser aplicada utilizando as posições e faces de uma pessoa de forma que seja feita uma comparação em uma base de dados.

Uma das principais características de um reconhecimento facial são os traços de uma pessoa, sendo eles chamados de pontos nodais, de forma que cada parte do rosto sejam identificados e aplicados em diversos tipos de processamentos e, conseqüentemente, são escolhidos de acordo com a necessidade e aplicação no sistema.

2.1 Conceitos Básicos

Diversos fatores são analisados para que seja feito um bom reconhecimento facial, e entre elas estão as seguintes características:

- a) Pose: As imagens analisadas para serem reconhecidas, podem sofrer variações devido à pose do indivíduo, tais como: Perfil, cabeça erguida e cabeça abaixada, podem ocorrer alterações no reconhecimento e os pontos focais irreconhecíveis como sobrancelhas, olhos, boca, etc;
- b) Expressão Facial: Alguns fatores que atrapalham em um reconhecimento como choro, risos, caretas, influenciam na face;
- c) Ambiente: A forma em que a imagem é apresentada pode dificultar um reconhecimento devido à ruídos ou até mesmo escassez/excesso de luminosidade;
- d) Rotação de Imagem: Imagens para ser identificadas, podem aparentar rotações horizontal e/ou vertical, fazendo com que a face fique distorcida dificultando o reconhecimento.

2.1.2 O que é reconhecimento facial

O reconhecimento facial automatizado é um conceito relativamente novo, desenvolvido pela primeira vez na década de 1960. Esta tecnologia chamou muito a atenção do público, quando em 2001, durante o SuperBowl da NFL (Liga Nacional de Futebol Americano), foram capturadas imagens de vigilância e comparadas com uma base de dados de foto-arquivos digitais (NUNES, 2015).

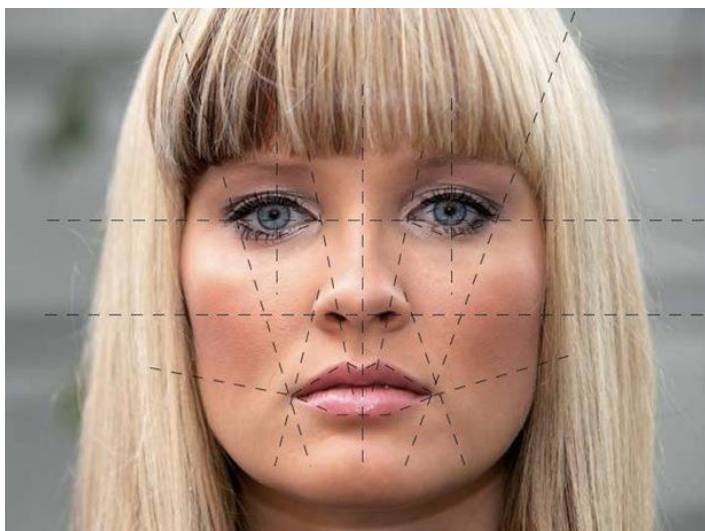
O conceito de identificação ou reconhecimento facial foi introduzido nos anos 1960. “Durante os anos 1964 e 1965 Woodrow Wilson Bledsoe, Helen Chan Wolf e Charles Bisson trabalharam no reconhecimento facial humano fazendo uso do computador e desenvolveram o primeiro sistema semiautomático de reconhecimento”. (GALVIS TRASLAVIÑA, 2015).

Um sistema de reconhecimento facial utiliza atributos exclusivos dos seres humanos. Esses sistemas procuram identificar características faciais únicas de cada pessoa, que possam ser medidos e utilizados como referências para o reconhecimento de quem a porta. Como por exemplo, a distância entre os olhos, sobrancelhas, etc. (PRAZERES, 2010).

O reconhecimento de face é uma particularização do reconhecimento de padrões, onde tem o objetivo de classificar informações baseados em conhecimento a priori ou em informações estatísticas dos padrões (MENEZES, 2009). Com base nisso, pode-se destacar o processo de reconhecimento tomando-se como base a imagem inicial contendo o rosto, sendo filtrada por um padrão definido, para que na saída apresente o reconhecimento, identificação e/ou categorização.

Esses dados são considerados como um conjunto de traços ou pode-se chamar de pontos nodais, onde vários locais no rosto são divididos para possíveis reconhecimentos e verificações. Além disso, a análise de cada ponto pode ser utilizada separadamente como, por exemplo: Utilizar apenas a parte dos olhos para verificar se uma pessoa está com eles fechados (dormindo, sonolento), utilizar a parte da boca e bochechas na análise de sentimentos caso está sorrindo, ou até mesmo as sobrancelhas caso a pessoa estiver brava. Abaixo na figura 2.1, é possível ver um exemplo dos pontos nodais e locais reconhecidos num rosto, que será seguido como base durante o projeto.

Figura 2.1 – Marcação de características da face humana



Fonte: TechTudo¹

2.1.3 Utilização

Os sistemas de reconhecimento facial têm por sua finalidade determinar a identidade da pessoa que está em um banco de dados prévio utilizando seus pontos nodais, viabilizando e melhorando processos em que é necessário ser feitos manualmente e, conseqüentemente, demanda mais tempo e esperas.

A utilização dessa tecnologia pode-se aplicar em uma gama de aplicações, entre elas estão: Controles de acesso, terminais de ônibus e aeroportos, bancos, universidades, entre outras. Considerando esses fatores, a melhoria principalmente em universidades se torna algo relevante devido ao grande fluxo de estudantes, professores e funcionários.

Para desenvolver uma aplicação de reconhecimento facial à partir da utilização de faces, é necessário uma série de treinamentos e métodos computacionais para que seja feito tal identificação. Atualmente vários tipos de algoritmos são estudados e criados para a melhoria tanto de desempenho, quanto ao grande número de dados e para isso várias técnicas são adotadas. Como dito por Osvaldo de Almeida (2006), as técnicas utilizadas são elas:

¹ Disponível em: <http://www.techtudo.com.br/artigos/noticia/2012/04/como-funciona-o-reconhecimento-facial.html>. Acesso em: 22 abr. 2017.

- a) Métodos baseados no conhecimento: Baseados no conhecimento humano para definir quais são as características que constituem a face. Normalmente essa técnica é utilizada para localização;
- b) Abordagem de características invariantes: Capacidade de encontrar e utilizar características em que não dependem do meio externo tais como, iluminação, variação de posição, entre outras. Semelhante a primeira técnica, também é utilizada para localização;
- c) Métodos de casamento de padrões (*Template Matching*): É um tipo de método que dentro de uma base de padrões existentes, é feita a comparação entre as duas imagens e o processo de validação é feito. Utilizados em localização e reconhecimento;
- d) Métodos baseados na aparência: Diferente do casamento de padrões e outros modelos, esse método é feito de forma em que o modelo é treinado à partir de um treinamento. Dessa maneira, o algoritmo aprende a identificar a face, nesse caso é o mais utilizado para reconhecimento.

2.2 Análise e Reconhecimento de Faces

Um dos grandes problemas ocorridos durante o processo de análise e reconhecimento facial é o fato de uma imagem/vídeo apresentar ruídos, mudança de posições, problemas de iluminação, modificações do rosto (tatuagens, barba, óculos, bonés, etc.). Esses aspectos afetam a validação e comparação no modelo treinado e com isso pode-se gerar valores de saída incorretos.

Como dito por Fernanda Nunes (2015), atualmente temos algoritmos dependentes de cenários e a classificação dos algoritmos de detecção de rostos pode ser: Métodos baseados em características faciais e métodos baseados na imagem.

As imagens abaixo mostram alguns fatores ocorridos que podem causar problemas no reconhecimento facial e possivelmente as análises podem ser comprometidas devido a serie de fatores encontrados durante o processo. A figura 2.2 exemplifica um problema no reconhecimento devido à tinta no rosto da torcedora.

Figura 2.2 – Problemas que podem afetar o reconhecimento facial



Fonte: Veja, 2010²

Outro exemplo que se pode apresentar são ruídos que ocorrem na imagem, e possivelmente em um vídeo afetaria a análise e identificação dos dados treinados. A figura 2.3 apresenta uma imagem com ruídos.

Figura 2.3 – Exemplo de imagens com desfoque e ruídos



Fonte: Tecmundo, 2014³

² Disponível em: <https://veja.abril.com.br/galeria-fotos/torcida-feminina/>. Acesso em: 22 abr. 2017.

³ Disponível em: <https://www.tecmundo.com.br/apps/64807-paquera-rosto-ideal-brasileiros-aplicativo-resposta.htm>. Acesso em: 22 abr. 2018.

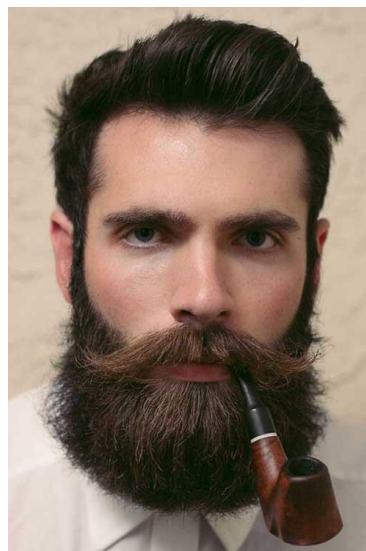
2.2.1 Características

A face humana possui inúmeras características que são utilizadas, de maneira intuitiva, para reconhecer as pessoas, como os olhos, sobrancelhas, boca, nariz e cabelo. Para descrever uma pessoa, é necessário analisar o posicionamento, forma e tamanho dessas características, além da distância que separa cada uma delas. A partir dessas informações é possível caracterizar uma pessoa (ALMEIDA, 2010).

Com essas informações, é possível aplicar em uma série de algoritmos e *frameworks* de processamento de imagens fazendo com que os dados recebidos na entrada sejam comparados com a base de treinamento e verificar se o reconhecimento foi feito de forma correta. Sendo assim, ao escolher um método que possibilita o reconhecimento e identificação da face, devem-se entender alguns pontos, como por exemplo: Modificações no rosto (barba, bigode, etc.), expressão facial (choro, risos, etc.) e até mesmo as condições do ambiente.

A figura 2.4 apresenta um rosto modificado utilizando barbas que afetariam para um bom reconhecimento facial.

Figura 2.4 – Exemplo de rosto com barba e bigode



Fonte: Barbearia, 2017⁴

⁴ Disponível em: <https://barbearia.org/barba-old-dutch/>. Acesso em: 22 abr. 2018.

Algumas características podem ser impedidas de reconhecer, e os pontos nodais rosto podem ser prejudicados pela barba e ocultar alguns pontos, porém com algumas ferramentas de processamento de imagens, é possível simular o rosto sem esses itens e possivelmente uma melhora no reconhecimento será apresentada.

2.2.2 Reconhecimento e identificação

Reconhecimento facial é uma tarefa simples para os humanos que até mesmo um bebê com poucos dias de vida já conseguem distinguir entre rostos de conhecidos e de desconhecidos. Enquanto isso, para os computadores este é um processo bem mais complexo e ainda não existe um sistema que se compare a eficiência do cérebro humano (NASCIMENTO, 2015).

Um dos grandes problemas na identificação de rostos é a detecção dos mesmos por meio de imagens. Para que o algoritmo funcione perfeitamente se deve fazer uma detecção precisa da imagem (NUNES, 2015).

Atualmente é possível utilizar vários algoritmos para que seja possível reconhecer e/ou identificar faces, e os métodos utilizados pode-se aplicar em diversos fatores, entre eles o treinamento de redes com dados salvos e pré-treinados ou então buscar características simples como olhos, sobrancelhas, lábios, etc.

Para fazer isso, é necessário delimitar algumas áreas desejadas para aplicar o procedimento de identificação e as dificuldades citadas nos itens anteriores dificultam os reconhecimentos devido a esses fatores. Porém, hoje em dia temos diversos algoritmos utilizados para detecção de faces, como por exemplo:

- a) Algoritmo de Viola-Jones;
- b) Algoritmo Padrão Binário Local (Local Binary Patterns);
- c) Eigenfaces.

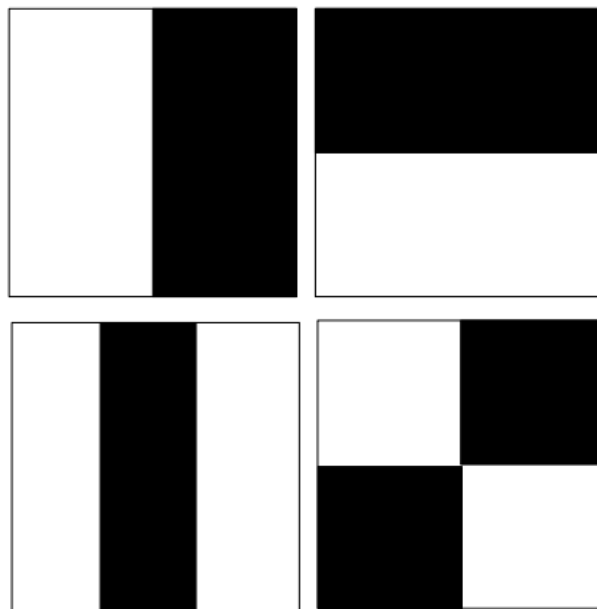
2.2.3 Algoritmos de reconhecimento

Hoje em dia, é possível dizer que houve um progresso muito grande na área de detecção e reconhecimento facial pelo fato do avanço da tecnologia para otimização de processos manuais.

Paul Viola e Michael Jones propôs em 2001 uma abordagem para detecção de objetos em imagens que se baseia em três conceitos: integral de imagem, treinamento de classificadores utilizando *boosting* e o uso de classificadores em cascata. Embora o algoritmo possa ser treinado para reconhecer qualquer objeto, a motivação principal da abordagem de Viola e Jones foi o reconhecimento facial. O ponto forte desse algoritmo é a rapidez com que é executado (BRAGA, 2013).

Os cálculos utilizando integral de imagem onde é possível determinar os padrões através de soma de áreas, que é aplicado a soma dos valores dos *pixels* de uma sub-região. Além disso, de acordo com Tulio Santos (2011), As unidades básicas do método Viola-Jones são os denominados *features* retangulares, essas *features* tem quatro tipos de configurações possíveis como demonstrado na figura 2.5:

Figura 2.5 – Quatro configurações possíveis de um *feature*



Fonte: Adaptado pelo autor, 2018

A equação abaixo determina como é calculado a integral dado uma coordenada:

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y') \quad (1)$$

onde $ii(x, y)$ é a integral da imagem, (x, y) são as coordenadas dos *pixels*, $i(x, y)$ é a imagem original, e $i(x', y')$ é o valor do *pixel* da imagem na coluna x' e y' . Após isso, é possível fazer o somatório dos *pixels* que ocupam uma determinada área desejada e assim é possível determinar os classificadores que são definidos pela função abaixo:

$$C(w) = \begin{cases} 1, & \text{se } \sum_{t=1}^T \alpha_t h_t(w) \leq \frac{1}{2} \sum_{t=1}^T h_t(w) \\ 0, & \text{caso contrário} \end{cases}; \text{ onde: } (2)$$

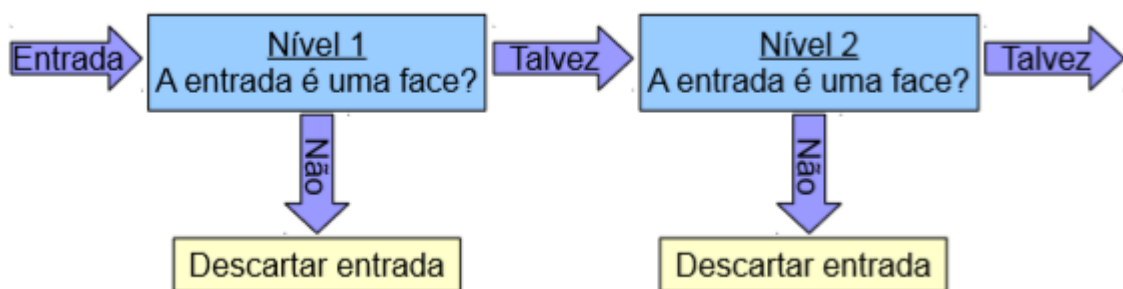
α_t : constante calculada durante o treinamento.

$h_t(w)$: valor do t -ésimo classificador fraco;

T : número de classificadores fracos;

Por fim, uma cascata de classificadores é utilizada, em que é passado uma entrada e ele classifica se é uma face ou não, e segue como uma cascata até encontrar realmente uma face. Caso uma entrada não for classificada como face, o algoritmo descarta a entrada e para, caso contrário, é passada para o próximo passo e todas as camadas devem ser percorridas para que o algoritmo seja finalizado, como mostra na figura 2.6:

Figura 2.6 – Representação Cascata de Classificadores



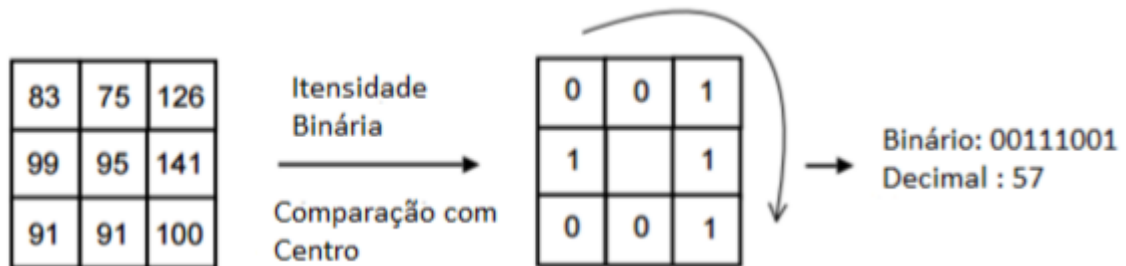
Fonte: Tulio Santos, 2011

Além do algoritmo de Viola Jones, é possível utilizar o algoritmo *Local Binary Patterns (LBP)*, que toma como base o algoritmo *Adaboost* para treinar os classificadores.

A utilização do LBP em uma imagem permite resumir a estrutura espacial de uma pequena parcela da imagem (8 *pixels*) em um único número (código LBP). Tal

código é definido a partir de uma vizinhança de 3x3 *pixels*, comparando-se com os *pixels* externos com o *pixel* central (NASCIMENTO, 2013). A representação do LBP está na figura 2.7.

Figura 2.7 – Representação da operação LBP

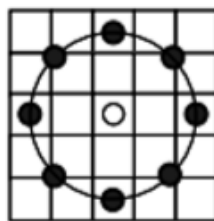


Fonte: NASCIMENTO, 2013

Na figura 2.7 foi apresentada a forma de obtenção do código LBP, onde é feito uma subtração do centro com os outros itens da matriz. Caso o valor subtraído for menor que zero, é atribuído o valor zero e caso contrário o valor um é adicionado na matriz. O preenchimento do centro não ocorre, pois é a base do sistema.

Após isso, o operador LBP foi estendido para utilizar círculos de diferentes raios para definir o código LBP, conforme apresentado na figura 2.8. Os pontos que não estão no centro dos *pixels* são definidos por interpolação.

Figura 2.8 – Operador LBP estendido.



Fonte: NASCIMENTO, 2013

Dessa forma, uma face é considerada a combinação de micro padrões que utilizando o algoritmo de LBP. Sendo assim, treinando esses classificadores com alguns algoritmos, pode-se definir um detector LBP que posteriormente, podem-se detectar faces.

3 PROCESSAMENTO DE IMAGENS

Como dito por Paula Ramos (2010), um dos objetivos de processamento de imagens consiste em melhorar o aspecto visual de certas feições estruturais, a fim de fornecer subsídios para que o analista humano possa realizar a interpretação da imagem de maneira confiável. Dessa maneira, pode-se utilizar essa técnica para diversas aplicações e objetivos, obtendo características ao fazer o processamento de forma automatizada.

As técnicas de processamento de imagens ultimamente tem um grande avanço principalmente na área de reconhecimentos faciais, de forma que a constante melhoria de tarefas que são manuais possa se tornar algo automático facilmente.

3.1 O que é Processamento de Imagens

O processamento de imagens envolve as técnicas de transformação de imagens, em que tanto a imagem original quanto a imagem resultado apresentam-se sob uma representação visual (geralmente matricial) (NETO, 2004). Utilizando como base esse conceito, pode-se dizer que as transformações procura melhorar as características visuais de uma determinada imagem ou *frame* utilizando as modificações como, por exemplo, alterações de brilho e contraste, redimensionamento (*resize*), ou até mesmo as correções de ruídos.

As alterações em uma imagem farão com que essa melhoria possa ser ajustada e reconhecida em diversas aplicações, tais como:

- Tratamento e melhorias das imagens recebidas para que seja reconhecido algum dado desejado pelo sistema, ou seja, com uma boa precisão nesse processamento é possível aperfeiçoar as entradas para obter uma saída desejada. As aplicações mais conhecidas utilizando desse conceito são elas: reconhecimento facial (extração de pontos nodais, correção de luminosidade de ambientes), imagens de satélites, controle de linhas de produção (defeitos de peças em sua criação), entre outras;

- Reconhecimento de modelos ou objetos localizados em alguma imagem. São aplicações mais utilizadas em: aeroportos (reconhecimento facial e extração de características para identificar suspeitos), automação em casas (abertura de portões/portas apenas com o rosto ou biometria salvo em um modelo treinado), ou até mesmo utilização na robótica para que faça o reconhecimento de um trajeto e não colida com objetos reconhecidos.

O processo de escolha da(s) técnica(s) de processamento depende de vários aspectos, mas os mais relevantes são as condições em que a imagem foi gerada, a qual pode ter iluminação inadequada, ruído, entre outros problemas que degradam a qualidade da imagem, que podem ser corrigidos via software, e as informações que se desejam extrair da imagem. (PEREIRA; PEREIRA, 2017).

3.2 Análise e Captura de Imagem

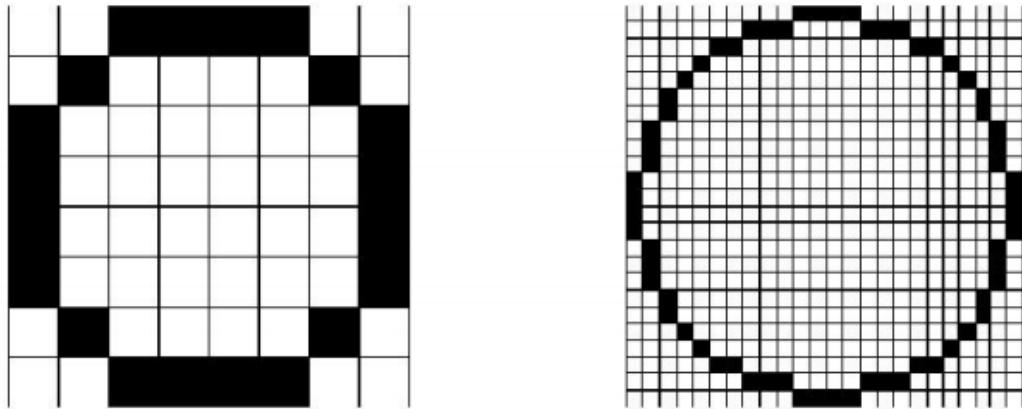
Conforme dito por Villegas (2009), as câmeras são recobertas por foto sensores, ou seja, sensores sensíveis a luz que são chamados de fotodiodos. Esses fotodiodos são capazes de gerar corrente elétrica diretamente proporcional a quantidade de luz que eles são atingidos. Uma matriz de fotodiodos é capaz de gerar uma imagem em tons de cinza.

Porém, quando é necessário fazer alguma alteração na imagem, é possível a sua manipulação de escalas, ao trocar de escalas de cinza para *Red Green Blue* (RGB), utilizando-se de diversas técnicas de pré-processamento, entre elas: saturação, brilho, contraste, etc. E após o momento de captura da imagem, é necessário modificar a imagem capturada a fim de facilitar a extração de informações da mesma, removendo as informações não desejadas da imagem ou destacando as informações necessárias para a fase de análise da imagem. (PEREIRA, 2018).

Algumas tarefas são necessárias para a melhora de um sistema utilizando processamento de imagens, como por exemplo, a alteração de tamanho para uma melhor precisão e desempenho na execução. Como dito por Antônio Lima (2009), resoluções maiores resultam em imagens com maior quantidade de *pixels*. Quanto maior a quantidade de *pixels* maior a complexidade computacional necessária para realizar tratamentos na imagem. Dessa forma é possível verificar a diferença na

relação entre uma imagem de alta resolução (maior número de *pixels*) e a outra com menor número de *pixels*, porém com poder computacional elevado como mostra a figura 3.1.

Figura 3.1 – Imagens com tamanhos iguais e resoluções diferentes



Fonte: SCURI, 2012

Ao criar sistemas computacionais para processar imagens estamos tentando simular o comportamento do sistema de visão biológico. Os computadores trabalharão no processamento das imagens assim como nosso cérebro faz com as imagens captadas pelos olhos (MANZI, 2007). Porém, esses sistemas necessitam de uma transformação das imagens reais para um formato que o computador reconheça que são as imagens digitais.

Como o processamento de um computador é capaz de manipular os *pixels* de forma muito precisa e alto desempenho, a utilização de processamento de imagens em larga escala para sistemas de reconhecimento facial é uma das melhores escolhas.

3.3 Técnicas de Processamento de Imagens

Um sistema de processamento de imagem eficaz deve ser aquele que seja capaz de identificar uma face e suas características através de uma determinada imagem, do mesmo modo que o ser humano identifica uma pessoa simplesmente

olhando para uma determinada fotografia, ou mesmo em uma transmissão de vídeo (NETO, 1997).

Para isso, existem diversas técnicas de processamento de imagens e é possível se trabalhar com ela de várias formas, entre elas, está o processamento *pixel a pixel*, ou reconhecimento de padrões que são pré-treinados (PEREIRA, 2017).

Dentre as variadas técnicas que são possíveis a sua utilização, e dentre elas, a de detecção de bordas (rostos) será a mais utilizada durante o trabalho, pois, é possível fazer diversas verificações em casos de alterações faciais possibilitando o desenvolvimento de um sistema preciso e que tenha capacidade de obter um alto índice de acertos.

Aplicações de procura por conteúdo presente na imagem em bases de dados de imagens faciais devem ser capazes de discernir entre uma grande quantidade de faces, e essas faces são sempre muito semelhantes entre si pelo fato de muitas delas apresentar contornos que são comuns. Dessa forma, existem técnicas no processamento digital de imagens que considera alguns procedimentos básicos para fazer os reconhecimentos, e nele estão contidos (NETO, 1997):

- Pré-processamento:
 - Segmentação para a utilização apenas dos rostos encontrados numa imagem, de forma que o sistema consiga obter um melhor desempenho por um tamanho menor de itens para processar;
 - Melhoria dos ambientes devido a problemas de iluminação, brilho, etc.
- Extração de características da Imagem facial, seguindo um modelo de representação de características consideradas discriminantes.

Dentre essas técnicas utilizadas, as de reconhecimento de padrões e a segmentação serão aplicadas no trabalho de forma que as faces sejam detectadas para posteriormente obter uma precisão no reconhecimento. A utilização das técnicas de processamento de imagens além de ter a capacidade de fazer o reconhecimento de padrões, ela pode ser utilizada para corrigir defeitos que a imagem possui, de forma que seja menos complexo o tratamento para uma execução com alto desempenho do algoritmo de reconhecimento facial.

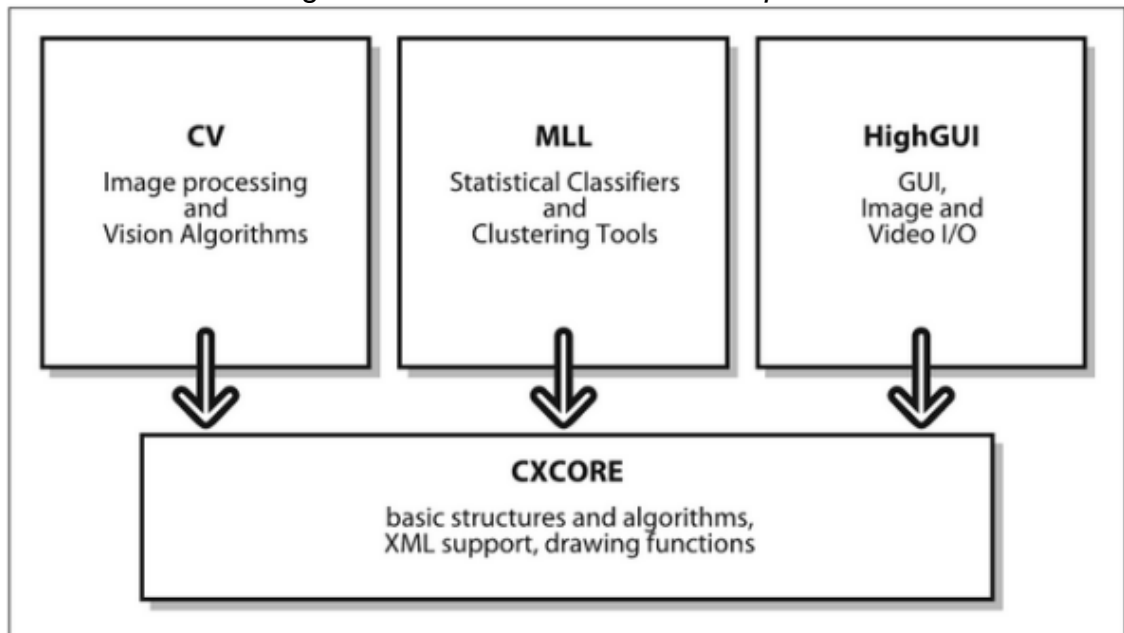
3.4 Opencv

OpenCV (Open Source Computer Vision Library) é uma biblioteca *open source* de licença *Berkeley Software Distribution* (BSD), disponível em C, C++, Java, Python, MacOS, iOS e *Android* que inclui centenas de algoritmos de visão computacional e possui suporte a aceleradores de hardware (NASCIMENTO, 2015).

Pode ser utilizada tanto para as plataformas Windows quanto Linux e, por esse motivo é muito utilizada para diversos *softwares* e, no entanto, como dito por Filipe Manzi (2007), entre as aplicações da biblioteca estão à identificação de objetos, segmentação e reconhecimento de imagens, reconhecimento de faces e gestos, captura de movimentos e reconhecimento de bordas.

A OpenCV visa prover as ferramentas básicas necessárias para encontrar a solução de um problema de visão computacional, e mesmo que as suas funcionalidades não sejam suficientes para casos complexos, os seus componentes básicos são suficientemente completos para auxiliar no desenvolvimento de uma solução concreta e completa (NUNES, 2016).

Ela é uma biblioteca muito completa, devido as suas funcionalidades atender diversas áreas de visão computacional, além de trazer para o desenvolvimento uma facilidade maior para se trabalhar com algoritmos de *Machine Learning* (classificações, análise de dados, entre outras), contêm estruturas de dados, tratamento de erros, interface gráfica de usuários (GUI), interface da câmera, e até mesmo possíveis rastreamentos de olhos e boca ou segmentação de fundo e frente que será muito utilizado no trabalho proposto. Alguns desses componentes citados formam a estrutura básica do *OpenCV* como mostrado na figura 3.2 abaixo:

Figura 3.2 – Estrutura básica do *OpenCV*

Fonte: Bradski; Kaehler, 2013

Baseado nesse conceito, as manipulações de imagem que a biblioteca oferece, traz uma segurança maior em ser utilizada pelo fator dela ser de código aberto, aplicável em multiplataformas e diversas funções de processamento de imagens serem capazes de trazer diversos benefícios em um desenvolvimento de sistemas com necessidades de processamento de imagens e reconhecimento de faces.

Dessa forma, depois de aplicado o processamento de imagens, o pré-processamento da imagem já tem sua forma que será utilizado na rede neural para aplicar nos rostos treinados e, conseqüentemente identificá-los através do reconhecimento facial.

4 REDES NEURAIS

As RNA (Redes Neurais Artificiais) surgiram da vontade do homem de reproduzir a máquina mais perfeita que existe até hoje, o cérebro humano (FALQUETO, 2007). A capacidade de executar tarefas com uma grande capacidade de raciocínio foi incorporada de forma que fosse possível uma resposta rápida e exata mesmo utilizando pouco recurso e dados.

Além disso, André de Carvalho, 2017 diz que essas redes são técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência onde se podem apresentar estruturas com centenas ou milhares de unidades de processamento.

4.1 Conceitos Básicos

“Uma RNA é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades geralmente são conectadas por canais de comunicação que estão associados a determinado peso, e a um conjunto de treinamento” (FALQUETO, 2007).

De acordo com Sergio Prazeres, 2010, essas unidades emulam, através de modelos matemáticos, o processamento de informações dos neurônios biológicos por meio de aprendizagem e aproximação. A interconexão desses elementos possibilita o processamento de informações em paralelo.

4.1.1 O que são redes neurais

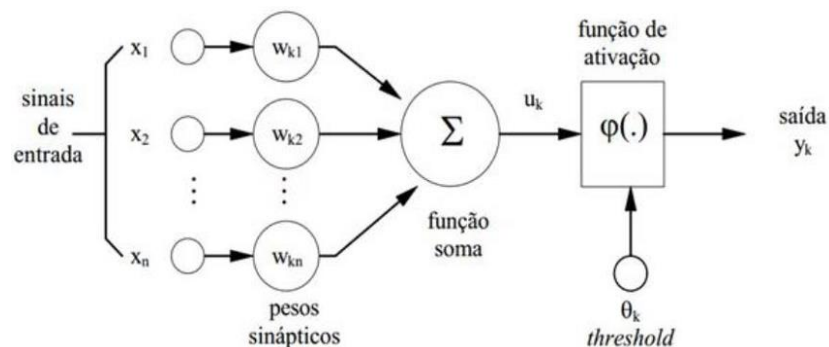
Uma rede neural é a interligação de unidades de processamentos, mais conhecidas como neurônios trabalhando de forma paralela, de tal forma que se pode identificar ou analisar uma estrutura. Esses neurônios podem estar designados em uma ou mais camadas utilizando junções geralmente unidirecionais.

Esses modelos existem a distribuição de pesos, onde eles armazenam o conhecimento adquirido conforme seus treinamentos e servem para aferir a entrada que cada neurônio recebe na rede. Como dito anteriormente, as RNAs tem como referência e inspiração o comportamento do cérebro humano. (PASSOS, 2017).

Tendo em vista essa teoria, pode-se dizer que é possível desenvolver um *software* que de certa forma, simule o mesmo comportamento de um cérebro humano para conseguir realizar diversas tarefas, tais como: aprender, detectar padrões, corrigir os erros, etc. Além disso, as redes neurais funcionam utilizando os neurônios que são mais conhecidos como *perceptrons* sendo eles os responsáveis por se basear em um comportamento humano para funcionar.

Nesse procedimento, é feito a análise da entrada de dados, as distribuições de pesos de modo que atenda a condição esperada. Como dito por Jordão Júnior, 2017, o corpo celular é composto por uma função de ativação e um potencial, semelhante ao potencial eletroquímico dos neurônios biológicos. Sendo assim, os neurônios artificiais utilizadas nas redes neurais tem uma grande semelhança com os biológicos e essa analogia está representada na figura 4.1.

Figura 4.1 – Modelo de um neurônio artificial



Fonte: HAYKIN, 2001

Imitar um cérebro humano não é uma tarefa fácil, porém ao utilizar as redes neurais, com esse poder de adaptação e aprendizado devido aos seus treinamentos, é possível desenvolver sistemas que após realizar os testes, se comporte de maneira coerente e possivelmente, possa ser embarcado em algum outro dispositivo com menor poder de processamento.

4.1.2 Treinamento das redes

“O treinamento de uma rede neural trata-se do ajuste dos parâmetros da rede por meio de estímulos ambientais conhecidos como padrões de dados de entrada ou de treinamento, a fim de melhorar seu desempenho” (ZUBEN, 2017).

Além disso, os treinamentos servem para definir um conjunto de regras que são adicionadas na sua rede de forma que é possível aumentar a sua capacidade de fazer melhores reconhecimentos e aprender de forma mais coerente quando é feito um ajuste dos pesos nas sinapses.

Existem dois tipos de treinamentos, e são eles: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço:

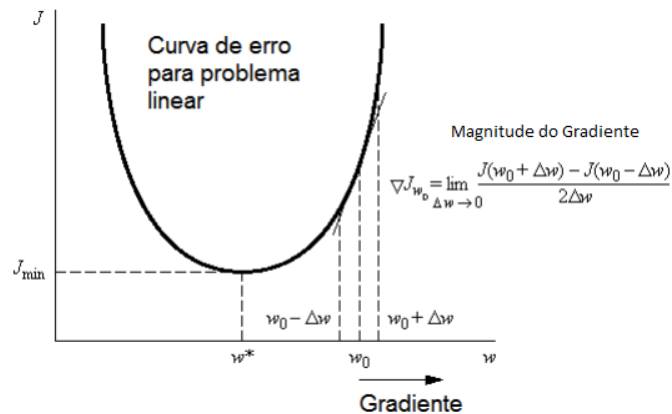
- a) Aprendizado supervisionado: Quando há uma resposta já pré-definida que pode ser comparado com o que a rede neural forneceu como saída para atingir os valores desejados;
- b) Aprendizado não supervisionado: Não existe um agente externo para avaliar a saída de acordo com os padrões de entrada da rede neural;
- c) Aprendizado por reforço: A resposta da rede é avaliada por um agente externo que define se o que foi retornado é coerente com o treinamento ou não.

Um dos algoritmos supervisionados mais conhecidos é o *backpropagation*, onde ele utiliza-se do método gradiente descendente, que faz a correção dos pesos no sentido contrário da propagação das informações (VERAS, 2013).

Surgido por volta de 1974, o algoritmo de *backpropagation*, destacou-se entre os outros que procuravam um valor w [$w_1, w_2, w_3 \dots w_n$] sendo eles um conjunto de pesos sinápticos para fazer a minimização dos erros, e faz com que as saídas que são comparadas tenha mais precisão pelo fato de ser um algoritmo supervisionado.

Como dito por Greco, 2011, no algoritmo *backpropagation*, busca-se o ponto da curva de erro onde o erro quadrático seja nulo e para isto os valores de w_i são ajustados de forma a deslocar o ponto da curva J sempre para o lado oposto do gradiente, até atingir valor nulo, e essa teoria é mostrada na Figura 4.2.

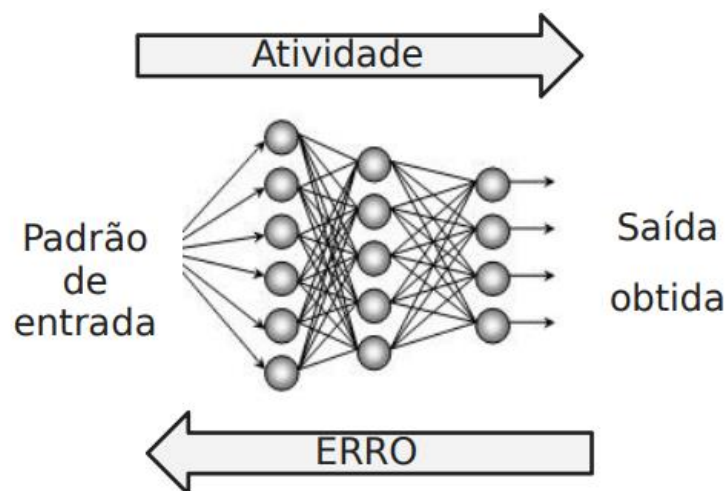
Figura 4.2 – Curva e gradiente do erro para problemas lineares



Fonte: GRECO, 2011

A propagação consiste na aquisição dos dados pela camada de entrada e sua propagação por toda rede, produzindo uma saída (SOUZA, 2010). Após esse procedimento como citado anteriormente, o algoritmo supervisionado faz uma comparação com um valor conhecido e a saída da rede. Caso haja alguma divergência nesses dados comparados, o processo de *backpropagation* se inicia e é retornado para fazer os reajustes dos pesos, e esse procedimento é feito até se atingir uma saída desejada. A figura 4.3 mostra uma representação desse algoritmo em funcionamento.

Figura 4.3 – Representação algoritmo *backpropagation*



Fonte: Adaptado de SOUZA, 2010

4.2 Tipos de Redes Neurais

A topologia das RNA's varia de acordo com a forma com que os neurônios se conectam, formando assim um conjunto (rede) de neurônios. Essa topologia pode ser chamada de redes *feedforward* ou de redes *feedback* (ZAMBIASI, 2017).

Além disso, as redes neurais se dividem de acordo com as suas características e diferentes tipos de aplicações. As redes mais conhecidas são elas: CNN (*Convolutional Neural Network*), DNN (*Deep Neural Network*), entre outras. Além disso, antes de entender sobre esses tipos de redes, é necessário falar sobre os neurônios artificiais, *perceptrons*, e as redes neurais artificiais (MLP – *Multilayer Perceptron*).

4.2.1 Neurônio artificial

O neurônio artificial é um modelo simplificado e simulado do neurônio real e suas características básicas são a adaptação e a representação de conhecimentos baseada em conexões (ZAMBIASI, 2017). Foi estudado sobre essa teoria e obteve-se uma conclusão de que o neurônio funciona de uma forma binária, ou seja, dependendo da sua entrada ele seria ativado ou não.

Utilizando a função de ativação pode-se definir se o neurônio será ativado ou não, e dessa forma é possível fazer com que um neurônio aprenda e forneça uma resposta precisa a ponto de ser similar com um biológico após passar por um processo de aprendizado. O modelo de um neurônio artificial pode ser visto anteriormente na figura 4.1.

4.2.2 Perceptron

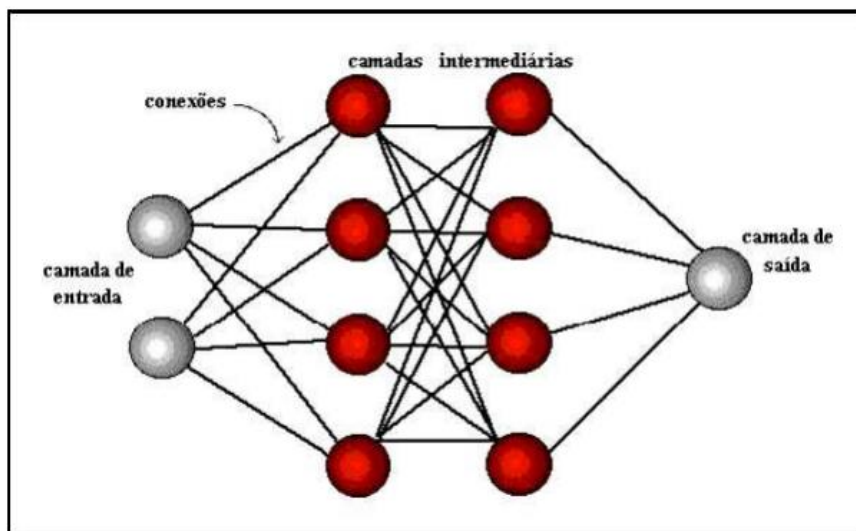
Frank Rosenblatt no final da década de 50, foi o responsável, induziu e desenvolveu a classe de RNA's chamada *perceptrons*, essa classe era uma rede genuína de múltiplos neurônios do tipo discriminadores lineares. (KOVÁCS, 2006).

O modelo de *perceptron* foi o primeiro a ser desenvolvido. Apresenta um conceito de neurônio artificial que ainda hoje é utilizado. Nesse modelo é feito uma

soma ponderada em cada um de seus neurônios e passa para uma função não-linear com limiarização (FALQUETO, 2007).

Essas redes são utilizadas em camadas e pode ser classificadas como: Camada de entrada, onde os padrões são apresentados a rede, a camada intermediária é responsável por fazer toda a parte do processamento através das conexões ponderadas, e a camada de saída define o resultado final concluído pelo processamento. A figura 4.4 mostra um modelo de camadas onde um *perceptron* possivelmente atuaria.

Figura 4.4 – Modelo de camadas RNA's



Fonte: FALQUETO, 2007

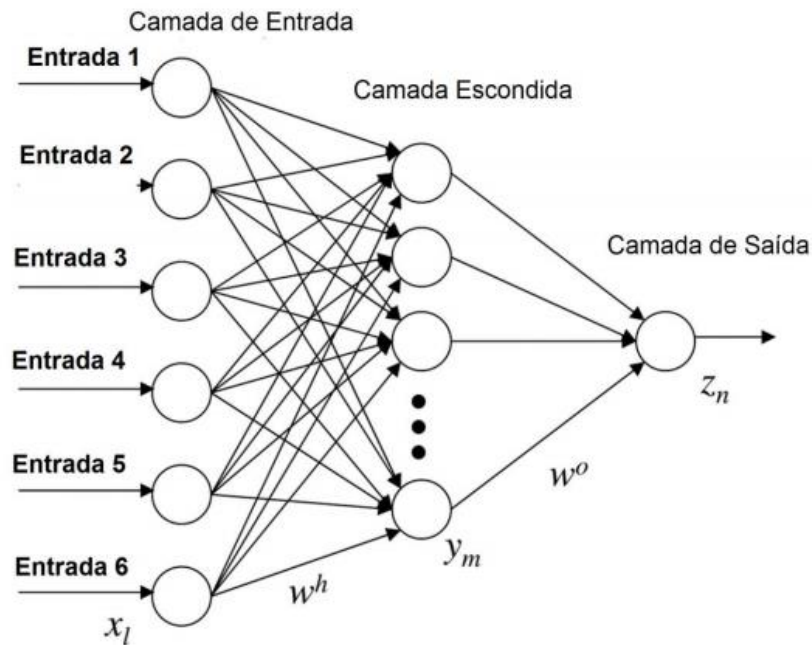
4.2.3 Redes neurais multicamadas perceptron (MLP)

Para esse trabalho, foi escolhida a *Multilayer Perceptron (MLP)*. Em um modelo onde não há processamento na camada de entrada mesmo se conectando com a camada de saída denominado de *perceptron* não era viável para o projeto, pois somente poderia solucionar problemas lineares.

É uma rede composta por uma camada de entrada, uma ou mais camadas intermediárias, e uma camada de saída. Em cada uma dessas camadas é possível conter um ou mais neurônios e elas estão dispostas de forma escondidas e além disso, pode ser utilizada de forma profunda, que são as chamadas “*Deep Neural*

Networks” (VERAS, 2013). A arquitetura de uma rede MLP pode ser analisada conforme mostra na Figura 4.5 abaixo:

Figura 4.5 – Estrutura de uma *Multilayer Perceptron*



Fonte: VERAS, 2013

Com essa teoria, é possível utilizar diversos tipos de redes neurais para aplicar em sistemas e utilizar-se de seus treinamentos e algoritmos de minimização de erros para que um problema seja resolvido e solucionado com maior precisão possível. Atualmente, existem diversas bibliotecas já compiladas que implementam métodos para facilitar o desenvolvimento das novas aplicações. Uma das bibliotecas que podem ser utilizadas para redes neurais é a D-lib, que será explicada na seção subsequente.

4.3 D-lib

A Dlib é uma biblioteca de *machine learning*, escrita em C++, que implementa diversos algoritmos e ferramentas para lidar com problemas dessa área, dentre eles, reconhecimento facial. Possui uma interface em *Python*, que é utilizada neste projeto

indiretamente, através da biblioteca *Face Recognition*, também em *Python* (DLIB, 2018).

Enquanto isso, o detector baseado em CNN é capaz de detectar rostos quase em todos os ângulos, tornando essa biblioteca poderosa para desenvolver aplicações que se utiliza o reconhecimento/detecção facial.

Além disso, a Dlib foi escolhida devido ao seu desempenho superior a biblioteca *Tensorflow* no reconhecimento facial, pois além dos testes com reconhecimento facial, a Dlib mostrou que mesmo com pouco recurso é possível manter um alto nível de reconhecimento junto com a sua velocidade de processamento da sua CNN.

5 DESENVOLVIMENTO

A proposta para esse projeto foi da utilização de uma plataforma em que era capturado o rosto do aluno por uma câmera e passado esse *frame* para a tarefa de pré-processamento da imagem.

Para isso, foi escolhida a criação de um sistema web em conjunto com uma câmera externa acoplada a uma máquina capturando o *frame* das imagens colhidas para ser executada no projeto. Após captura-lo onde nele estão as pessoas para ser reconhecidas, é utilizada a biblioteca Dlib com a sua CNN para detectar as informações dos rostos encontrados.

O projeto é composto de três etapas para o reconhecimento das faces encontradas na imagem, sendo elas: captura do *frame* e tratamento da imagem, processamento e reconhecimento dos rostos e a verificação das informações dos rostos com a base de imagens conhecidas.

Após a execução dessas etapas, o sistema apresentará no próprio vídeo da câmera logo abaixo da face, o seu RA (Registro do Aluno) condizente à imagem capturada no momento da matrícula e assim sua presença é computada.

5.1 Arquitetura da Aplicação

A câmera do projeto a princípio foi utilizada a integrada de um *MacBook Pro* e esta possui uma ótima resolução para utilizar no reconhecimento de faces. Em sistemas para reconhecimento facial, a importância de uma imagem em boa resolução é grande, pelo fato de ser possível identificar mais facilmente e possivelmente reduzir falhas por problemas nas câmeras.

Porém, as imagens com maiores resoluções, requer um elevado poder computacional para processá-las, pelo seu maior número de *pixels* e para o nosso projeto possivelmente não atenderia a proposta inicial. Para contornar esse problema, foi decidido utilizar uma câmera externa (*Webcam*), mesmo com uma resolução mais baixa, e dessa forma foi possível utilizá-la no sistema. Após fazer o *resize*, será chamada a biblioteca com a rede convolucional para identificar os rostos

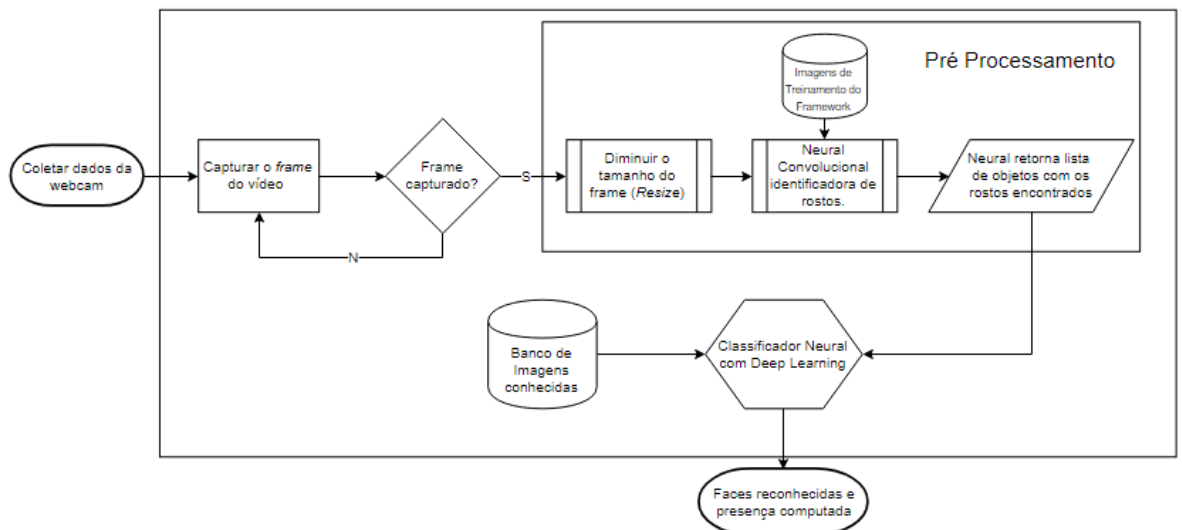
presentes no *frame* para fazer a comparação entre o *dataset* de imagens conhecidas com a imagem atual utilizando o classificador neural.

A aplicação será executada em uma plataforma *web*, utilizando a *framework* *Flask* em conjunto com a linguagem de programação *Python*, e essa linguagem foi escolhida por conta da CNN e o classificador neural serem desenvolvidos com ela, torna-se o sistema com maior facilidade de manutenção além de conter muitas ferramentas disponíveis para utilização.

Flask é um “*microframework*” destinado, principalmente, a pequenas aplicações com requisitos mais simples, aborda uma teoria onde os desenvolvedores que escolhem a forma que seus dados sejam armazenados, além de ser muito utilizado em projetos menores (GARBADE, 2017).

O projeto contém um banco de imagens onde estão todos os alunos de uma determinada turma para ser comparada com os dados identificados pela Dlib ao entrar no classificador neural. Esse classificador contém uma rede neural profunda que se baseia no modelo *ResNet* (*Residual Network*) com 50 camadas, onde compara todos os pontos nodais trazidos pela CNN com as imagens no banco. A forma como o processo de reconhecimento é feito, é apresentado na figura 5.1.

Figura 5.1 – Etapas do sistema reconhecedor de faces



Fonte: Elaborado pelos autores

5.2 Treinamento da Rede Neural

Durante o desenvolvimento do projeto, foi necessário fazer o treinamento da rede neural profunda, e dividiu-se esse método em duas etapas:

Análise e identificação das imagens conhecidas: Coleta de dados da câmera, aplicar a redução do tamanho (*resize*) para melhor processamento dos dados e desempenho da rede, além da execução das imagens na CNN da biblioteca Dlib para identificar as faces encontradas e retornar os objetos complexos para a entrada na rede neural profunda;

Treinamento do classificador neural: Definição dos objetos para realizar o treinamento do classificador, bem como a execução dos métodos de treinamento e suas épocas, além da explicação de como é feito a comparação com a base de imagens conhecidas e a resposta, se uma face é reconhecida ou não.

5.2.1 Análise e identificação das imagens conhecidas

Primeiramente, foi inicializado o projeto e capturar os dados de vídeo pela *webcam* para ser utilizado no sistema. Feito isso, identificou-se a necessidade de separar um *frame* para ser executado o pré-processamento na Dlib. Essa captura dos dados da câmera e como a imagem fica disposta na aplicação está exemplificada na figura 5.2.

Figura 5.2 – Captura do *frame* durante a execução do sistema



Fonte: Elaborada pelos autores

Começando a fase de pré-processamento, logo após esse procedimento de capturar a imagem, foi necessário fazer um ajuste no tamanho da mesma, pois a utilização da rede neural com o menor número de *pixels* torna o seu desempenho muito melhor, e dessa forma é possível ganhar capacidade de processamento. A figura 5.3 mostra o trecho de código que aplica a rotina de *resize* antes de entrar no método da rede convolucional.

Figura 5.3 – Rotina aplicando redução de tamanho da imagem (*resize*)

```
unknown_image = face_recognition.load_image_file(image_to_check)

# Diminuir o tamanho da imagem (resize) para melhor desempenho do sistema.
if max(unknown_image.shape) > 1600:
    pil_img = PIL.Image.fromarray(unknown_image)
    pil_img.thumbnail((1600, 1600), PIL.Image.LANCZOS)
    unknown_image = np.array(pil_img)
```

Fonte: Elaborada pelos autores

Como citado anteriormente, durante o desenvolvimento foi utilizada a biblioteca Dlib para identificar os rostos existentes no *frame* e fazer a conversão das informações da face em um objeto complexo, ou seja, transformar em uma matriz de números as características de cada ponto nodal. A figura 5.4 mostra um exemplo de uma imagem após passar pelo processo de identificação da CNN e a figura 5.5 apresenta uma parte da matriz de 128 dimensões que faz a conversão dos rostos em números para entrar no classificador neural.

Figura 5.4 – Características de uma face utilizando Dlib



Fonte: Elaborada pelos autores

Figura 5.5 – Matriz de valores dos pontos identificados no rosto

```
[array([ 0.02207182,  0.02207182,  0.02207182,  0.02207182,  0.02207182,
        -0.20207794,  0.0921451 ,  0.11184137, -0.00764854, -0.08805326,
         0.01475809, -0.02373593, -0.03956413,  0.12056189, -0.01607963,
         0.24416386, -0.05227867, -0.25743023, -0.07824387, -0.01248436,
         0.16125533, -0.10774027, -0.12126105, -0.01941909, -0.04988828,
         0.11296619,  0.06437065, -0.04557081,  0.07520494, -0.154365 ,
        -0.35348928,  0.01738055, -0.10761245, -0.07989901, -0.11699589,
        -0.08177621,  0.05825568, -0.13615549, -0.0676811 ,  0.00891522,
         0.16760695, -0.01277129, -0.02049825,  0.10235873, -0.02882695,
        -0.19271691,  0.00724913,  0.07781079,  0.28189442,  0.14359334,
         0.07065558,  0.03708972, -0.03107995,  0.12073345, -0.32213145,
         0.04657434,  0.09980447,  0.12616335,  0.01704407,  0.08365649,
        -0.16482928, -0.04135096,  0.09174224, -0.16956919,  0.12514792,
         0.03221683, -0.11390231, -0.0378969 , -0.09249835,  0.15843979,
         0.09909201, -0.13230872, -0.07265665,  0.06552565, -0.14342283,
        -0.1481103 ,  0.04497981, -0.07844096, -0.23157512, -0.36257964,
         0.05123711,  0.33867627,  0.12561025, -0.25938398,  0.03065751,
         0.02766028,  0.04174296,  0.08368013,  0.11868128, -0.0250628 ,
         0.05889636, -0.16890228, -0.00125279,  0.19505185, -0.00635442,
         0.00375431,  0.2565496 ,  0.0110358 ,  0.05635849,  0.031846 ,
         0.14723775, -0.07613221, -0.01632117, -0.11317981, -0.03344366,
         0.11117581, -0.0583885 , -0.00430694,  0.07455104, -0.22706105,
         0.13980904, -0.06429489,  0.0437548 , -0.00509384, -0.03250811,
        -0.05812979,  0.02886757,  0.13263944, -0.22668828,  0.11659954,
         0.21487111, -0.01244355,  0.16255561,  0.08482601, -0.00904587,
         0.02084307, -0.03724389, -0.18526608, -0.11086601,  0.06702941,
        -0.04075371,  0.02579562,  0.06846698]])]
```

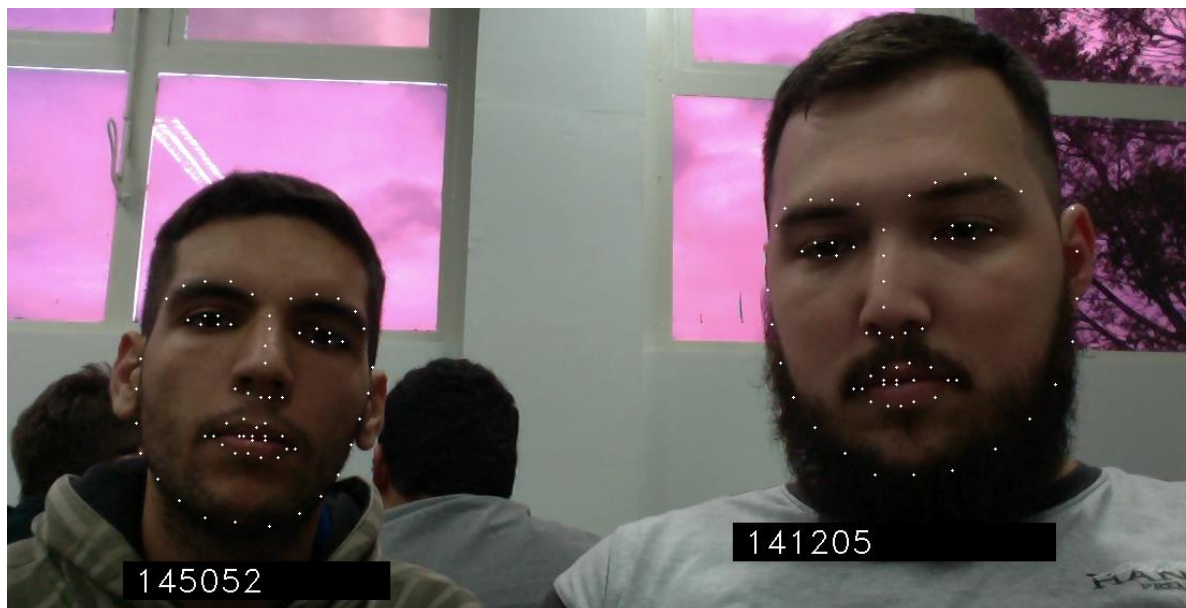
Fonte: Elaborada pelos autores

5.2.2 Treinamento do classificador neural

Para ser feito o treinamento do classificador neural que é uma rede neural profunda (DNN), foram utilizados alguns modelos, sendo eles: Um modelo de 68 pontos, outro de cinco pontos, a resposta da CNN e o *face encoder*.

Feito isso, existe uma função que contém um *loop* onde pega a imagem de todos os alunos daquela turma na base conhecida, processa esse *frame* para que a informação colocada no classificador neural seja condizente com o que vem do pré-processamento feito pela Dlib. Esse laço de repetição contém uma validação e processa face a face apresentadas, pois durante a captura do vídeo pode ser que apareça um ou mais alunos e todos eles deverão ser identificados. A figura 5.6 mostra um reconhecimento feito com mais de uma pessoa, e a forma que o sistema se comporta em executar o reconhecimento.

Figura 5.6 – Reconhecimento com mais de uma pessoa



Fonte: Elaborada pelos autores

Caso durante esse processo de treinamento e feito ele, o sistema parte para o reconhecimento, e se a imagem coincide com uma das cadastradas para a turma no banco de imagens, o sistema identificará seu rosto e apresentará o registro do aluno(s) e a presença é computada e salva em um arquivo texto todos os RA's que o sistema identificou na sua execução. A figura 5.7 mostra um aluno reconhecido pelo

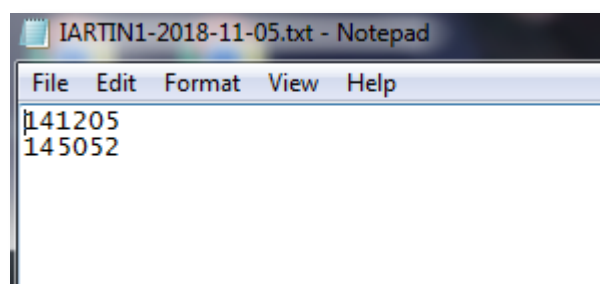
sistema e a figura 5.8 mostra o arquivo com os RA's identificados. Caso não seja possível realizar o reconhecimento de um aluno específico, será apresentado na tela o *label* “DESCONHECIDO” e não computará presença para ele, nesse caso, o aluno deverá comunicar com o professor para verificar e aplicar a presença manualmente.

Figura 5.7 – Aluno reconhecido pelo sistema



Fonte: Elaborada pelos autores

Figura 5.8 – Arquivo texto com os alunos reconhecidos



Fonte: Elaborada pelos autores

5.3 Resultados

O processo de validação em um sistema de reconhecimento facial é algo complexo devido aos problemas de alterações nas faces e a chance de erro deve ser mínima de forma que possa ser garantido que o sistema apresente um bom desempenho e alta taxa de acertos. Para isso, foi pensado em uma forma de integrar os dados colhidos e treinados e o desenvolvimento otimizado para uma aplicação que caso houver uma maior escalabilidade, não ofereça gargalos ou quedas de desempenho devido ao número massivo de dados e, consequentemente não ser viável sua utilização.

Após a instalação e configuração das ferramentas utilizadas no desenvolvimento do projeto, foi integrado as imagens conhecidas com a rede neural convolucional e o classificador neural, assim pode-se iniciar a fase de testes da aplicação e validações que serão discutidos nas seções subsequentes.

5.3.1 Validação

Com a CNN (*Convolutional Neural Network*) e o classificador neural implementado, foi iniciado a rotina de testes e validações do sistema, e a partir de um *dataset* de aproximadamente 35 alunos, obteve-se um bom desempenho, pois foram analisados cinco *frames* para cada aluno e a taxa de *frames* reconhecidos ficou acima de 60% na maioria dos casos, salvo exceções de alterações extremas e fotos não atualizadas para computar a presença.

5.3.2 Comparação com o *dataset*

Foi feito comparações com o *dataset* e conforme dito na seção 5.3.1, as alterações extremas na face não tem um bom comportamento no sistema, porém manteve um *rating* satisfatório e com um desempenho muito agradável, mostrou-se apto para utilizar grande quantidade de dados para seu treinamento e reconhecimento. A tabela 5.1 demonstra o resultado de um teste que foi feito,

considerando as imagens disponibilizadas dos alunos e a validação do algoritmo e classificação dos *frames* capturados.

Tabela 5.1 – Resultados da validação

Quantidade Alunos	35
<i>Frames</i> Analisados	170
<i>Frames</i> Reconhecidos	125
Taxa Reconhecida (%)	73,529

Fonte: Elaborado pelos autores

Além disso, para os dados em que obtiveram a menor taxa como mostrado em vermelho na Tabela 5.2, as fotos que não haviam sido atualizadas pelos alunos e houve mudanças na face como, por exemplo: barbas, óculos, etc. Para isso, o sistema mostrou a taxa de 40% no *rating* quando apresentou dificuldades durante o reconhecimento facial e não conseguiu reconhecer a maioria das imagens.

Tabela 5.2 – Resultados com taxa de acertos (*Rating*)

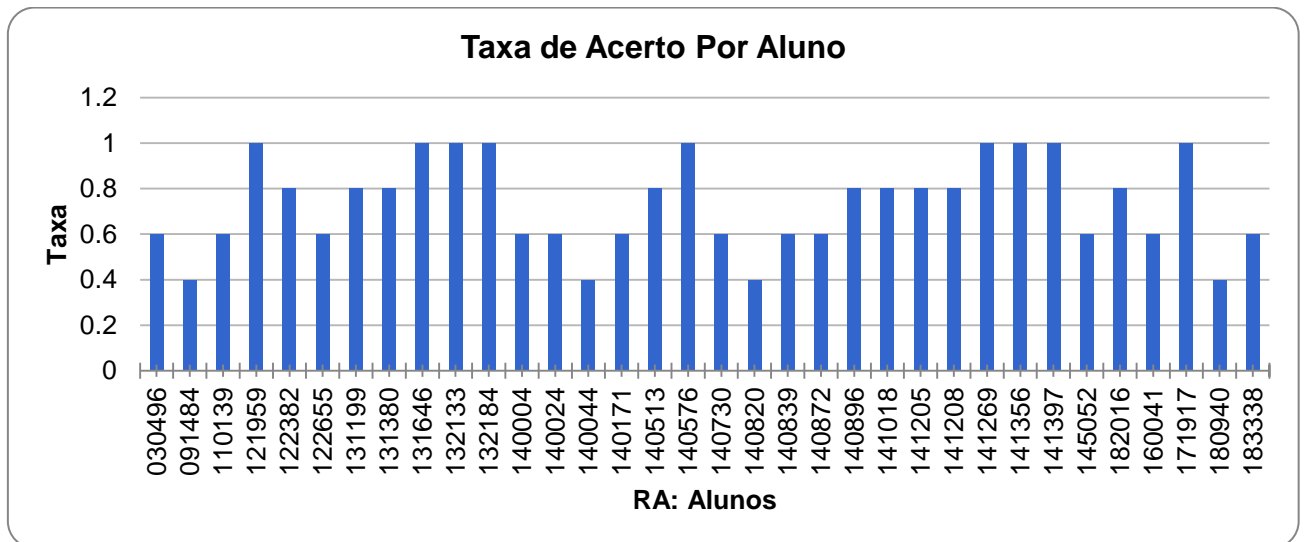
RA	Quantidade de <i>frames</i> analisados	Quantidade de <i>frames</i> reconhecidos	<i>Rating</i>
030496	5	3	0.6
091484	5	2	0.4
110139	5	3	0.6
121959	5	5	1
122382	5	4	0.8

Fonte: Elaborado pelos autores

Após o processo de validação dos dados, verificou-se através de gráficos e tabelas geradas pela aplicação que, durante o reconhecimento, manteve-se uma

estabilidade e não apresentou travamentos e mostrou taxa máxima para nove alunos e a maioria entre 60 e 80% dos *frames* reconhecidos. A figura 5.9 mostra o gráfico de todos os *frames* analisados e o comportamento ao reconhecer os cinco *frames* de cada aluno para uma sala de 35 pessoas.

Figura 5.9 – Gráfico da taxa de acerto x alunos



Fonte: Elaborado pelos autores

5.3.3 Taxa de acerto do sistema

Com base nos resultados dos testes executados após o desenvolvimento da aplicação, pode-se concluir que o sistema apresenta uma taxa de acerto acima da média tornando-o viável para utilização tanto em salas com número reduzido de alunos, quanto uma sala com maior quantidade de pessoas sem que haja algum tempo de gargalo ou queda de rendimento. Todos os testes apresentaram resultados positivos, satisfatórios, reconhecendo as pessoas mesmo com pequenas alterações no rosto. O maior problema encontrado para fazer o reconhecimento foi a verificação lateral da face, onde houve maior índice de erros e a *tag* desconhecido era atribuído ao aluno.

6 CONCLUSÃO

O algoritmo proposto e desenvolvido visou realizar o reconhecimento facial aplicado para um sistema de controle de presenças, e dessa forma pode-se concluir que a solução é de suma importância no ambiente acadêmico, pois faz com que um processo manual seja melhorado tanto para o professor, quanto para o aluno. Para tal feito, foram aplicados as técnicas de redes neurais e aprendizado profundo além de uma base de dados de testes de aproximadamente 50 faces para serem reconhecidas na execução da aplicação.

A aplicação das técnicas de reconhecimento facial utilizando a biblioteca D-Lib apresentou resultados satisfatórios, com baixa taxa de erros chegando até 95% a precisão em uma média de 30 alunos de uma disciplina. Entretanto, essa biblioteca mostra que mesmo não sendo muito conhecida no mercado, ela tem muito a crescer nessa área de reconhecimento facial. Sendo assim, sua utilização para extrair características do rosto com a sua CNN foi essencial para o sucesso do projeto, e através da junção de todos esses pontos citados acima durante o desenvolvimento do projeto e a programação do classificador neural utilizando uma rede neural profunda, pôde-se comprovar que com um hardware de baixo poder computacional, recurso reduzidos, e apenas uma foto para treinamento da rede neural, é possível fazer um reconhecimento facial rápido e preciso.

Os objetivos desse trabalho foram atingidos. Uma aplicação capaz de fazer o reconhecimento facial dos alunos para aplicar a presença e substituir o método não automatizado foi desenvolvida. As taxas de acerto do sistema comprovam que o mesmo é capaz de realizar todo o controle de presenças de forma agradável, veloz e com uma interface de usuário que mostre a utilidade do sistema como um todo.

Futuramente, o projeto pode ser embarcado em um dispositivo como, por exemplo, um *tablet* e ser inserido na porta das salas de aula, para que seja computada uma chamada totalmente automática, de forma que o próprio aluno ao entrar em sua sala aproxime a face, é computada a entrada, e ao sair da sala, realize o mesmo procedimento aplicando a presença por completo.

REFERÊNCIAS

- ALMEIDA, O. C. **Técnicas de processamento de imagens para localização e reconhecimento de faces**. 2006. 110 f. Trabalho de Conclusão de Curso (Mestrado em Ciência da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP. 2006.
- BRADSKI, G.; KAEHLER, A. ***Learning OpenCV: Computer vision with the OpenCV library***. 1.ed. O'Reilly Media, Inc., 2008. 560p.
- BRAGA, L. F. Z. **Sistema de reconhecimento facial**. 2013. 84 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP. 2013.
- CARVALHO, A. P. de L. F. **Redes neurais artificiais**. Disponível em: <<http://conteudo.icmc.usp.br/pessoas/andre/research/neural/>>. Acesso em: 24 out. 2018.
- DEVI S. S; MANE, P. K; AJAYKUMAR D. **“Face Detection System using OpenCV on Beagle Board”**
- DLIB **C++ Library**. 2018. Disponível em: <<http://dlib.net>>. Acesso em: 28 out. 2018.
- FALQUETO, D. **Rede Neural Artificial para Reconhecimento de Tabela de Horários de Arme/Desarme no Sistema Sigma**. 2007. 54 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Vale do Itajaí, Centro de Ciências Tecnológicas da Terra e do Mar, São José, SC. 2007.
- GARBADE, M. J. **Django, Flask ou Pyramid: Qual é o melhor framework Python para você?**. Georgia, 2017. Disponível em: <<https://blog.liveedu.tv/django-flask-pyramid-framework-python/>>. Acesso em: 31 out. 2018.
- GRECO, F. S. **Redes Neurais para Reconhecimento de Padrões de Vídeo MPEG4**. 2011. 37f. Trabalho de Conclusão de Curso (Engenharia Elétrica) – Universidade Federal do Paraná, UFP, Curitiba, PR. 2011.
- HAYKIN, S. **Redes Neurais: Princípios e Prática**. 2.ed. Bookman, 2001. 908p.
- KOVÁCS, Z. L. **Redes Neurais Artificiais: Fundamentos e Aplicações**. 4.ed. Livraria da Física, 2006. 42p.
- MANZI, F. A. **Aplicação de Visão Computacional para Extração de Características em Imagens do Olho Humano**. 2007. 47f. Trabalho de Conclusão de Curso (Engenharia da Computação com ênfase em Sistemas Embarcados) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, SP. 2007.

MATTOS, G. C. **PresentEye: Sistema de Controle de Presença por Reconhecimento Facial**. 2017. 72 f. Trabalho de Conclusão de Curso (Sistemas de Informação) – Escola de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro, UNIRIO, Rio de Janeiro, RJ. 2017.

NASCIMENTO, V. **Implementação de um sistema de identificação facial utilizando Linux Embarcado**. 2015. 85 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP. 2015.

NATAL de OLIVEIRA JUNIOR, J. **Uso de Rede Neural *Perceptron* Multicamadas como estimador de tempos de Flicker**. 2017. 48 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica com ênfase em Eletrônica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP. 2017.

NETO, C. R. **Processamento Digital de Imagens. Implementação de Watermarking**. 2004. 74 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação) – Faculdade de Ciências Exatas e Tecnologia, Centro Universitário de Brasília, Brasília, DF. 2004.

NETO E. L. A. – **Sistemas de Identificação Pessoal Utilizando Técnicas de Reconhecimento e Verificação Facial Automáticas**, 1997. 137 f. Mestrado em Engenharia Elétrica – Universidade Estadual de Campinas, Campinas, SP. 1997.

NUNES, F. T. **Técnicas de biometria baseadas em padrões faciais e sua utilização na segurança pública**. 2015. 65f. Trabalho de Conclusão de Curso (Pós Graduação em Tecnologias da Informação e Comunicação Aplicada a Segurança Pública e Direitos Humanos) – Univesidade Federal de Santa Catarina, Santa Catarina, SC. 2015.

NUNES, L. F. M.. **Reconhecimento Facial Biométrico Em Nuvens de Pontos Tridimensionais**. 2016. 111 f. Trabalho de Graduação em Engenharia de Controle e Automação – Universidade de Brasília, UnB, Brasília, DF. 2016.

PASSOS, I. **Uso de Redes Neurais Artificiais para a Detecção das Doenças Olho de Boi e Mancha de Sarna em Maçãs**. 2017. 61 f. Trabalho de Graduação em Ciência da Computação – Universidade de Caxias do Sul, UCS, Caxias do Sul, RS. 2017.

PEREIRA, R. L. C.; PEREIRA, F. S. **Reconhecimento de Libras com CNTK e Realsense**. 2018. 64 f. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) – Faculdade de Engenharia de Sorocaba, FACENS, Sorocaba, SP. 2018.

PISA, P. **Como funciona o RECONHECIMENTO FACIAL**. Disponível em: < <http://www.techtudo.com.br/artigos/noticia/2012/04/como-funciona-o-reconhecimento-facial.html> >. Acesso em: 22 mar. 2018.

SIQUEIRA, C. **Comissão aprova frequência mínima de 75% às aulas para ensino superior**. Disponível em: <

<http://www2.camara.leg.br/camaranoticias/noticias/EDUCACAO-E-CULTURA/432050-COMISSAO-APROVA-FREQUENCIA-MINIMA-DE-75-AS-AULAS-PARA-ENSINO-SUPERIOR.html>>. Acesso em 28 out. 2018.

SCURI, A. E. **Fundamentos da imagem digital**. Rio de Janeiro, 2002. Disponível em: <<http://webserver2.tecgraf.puc-rio.br/~scuri/download/fid.pdf>>. Acesso em: 20 ago. 2018.

VERAS, C. V. A. **Estudo Comparativo de Técnicas de Redes Neurais Artificiais na Previsão da Velocidade do Vento em Curto Prazo**. 2013. 57 f. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) – Universidade de Pernambuco, Escola Politécnica de Pernambuco, Recife, PE. 2013.

ZAMBIASI, S. P., **O neurônio artificial**. Disponível em: <http://www.gsigma.ufsc.br/~popov/aulas/rna/neuronio_artificial/index.html>. Acesso em: 28 out. 2018.

ZUBEN, J. V.; CASTRO L. N., **Redes neurais artificiais**. Disponível em: <ftp://vm1-dca.fee.unicamp.br/pub/docs/vonzuben/ia006_03/topico5_03.pdf>. Acesso em: 20 out. 2018.