

UNIVERSIDADE DE TAUBATÉ

**ÁLEF LUCAS PEREIRA NUNES
PEDRO MIRANDA SANTOS BUENO DOS REIS**

Programa de reconhecimento facial

**Taubaté – SP
Novembro de 2019**

**ÁLEF LUCAS PEREIRA NUNES
PEDRO MIRANDA SANTOS BUENO DOS REIS**

Programa de reconhecimento facial

Monografia apresentada para obtenção do certificado de graduação pelo curso de Engenharia da Computação da Universidade de Taubaté, Orientador: Edgar Israel.

Universidade de Taubaté – UNITAU

Engenharia da Computação

**Taubaté - SP
Novembro de 2019**

Sistema Integrado de Bibliotecas – SIBi / UNITAU
Biblioteca Setorial do Departamento de Informática

N972p Nunes, Álef Lucas Pereira
Programa de reconhecimento facial / Álef Lucas Pereira Nunes, Pedro
Miranda Santos Bueno dos Reis. - 2019.
40f. : il.

Monografia (graduação) - Universidade de Taubaté, Departamento de
Informática, 2019.
Orientação: Prof. Edgar Israel, Departamento de Informática.

1. Identificação de imagens. 2. *Deep learning*. 3. Visão computacional.
I. Reis, Pedro Miranda Santos Bueno dos. II. Universidade Taubaté.
III. Título.

CDD 005.1

Agradecimentos

Temos o nosso agradecimento direcionado a todos que contribuíram para que a realização deste trabalho se concretiza-se. Primeiramente, a Deus, pois sem Ele nada seria possível, a todos professores da Universidade de Taubaté que nos conduziram durante toda nossa graduação, guiando-nos através da passagem de todo o conhecimento que carregaremos em nossa vida profissional e pessoal, aos nossos colegas, amigos e familiares que sempre estiveram do nosso lado e nos apoiaram a seguir o nosso grande sonho. Finalmente, agradecemos ao nosso orientador Edgar Israel, por estar disponível para as devidas orientações e dando-nos o suporte necessário para realização deste trabalho.

“Success consists of going from failure
to failure without loss of enthusiasm.”

Winston Churchill

Resumo

O reconhecimento de imagens está a cada dia mais presente em nossa rotina, e com esse avanço temos o uso constante da biometria para garantir a segurança de acesso e identificação, dessa forma tecnologias como o reconhecimento biométrico de digital, fala e o reconhecimento facial estão sendo amplamente utilizados, tornando assim a pesquisa e desenvolvimento do tema algo vital para o avanço da tecnologia. Neste trabalho foi demonstrado o desenvolvimento de um programa de reconhecimento facial utilizando ferramentas de inteligência artificial como aprendizado de máquina e deep learning com o objetivo de realizar a correta identificação de faces em fotos e os reconhecimentos das mesmas, diferenciando-as umas das outras. Foi observado que o programa desenvolvido tem uma boa assertividade, mesmo com o uso de imagens com baixa resolução e em ângulos pouco favoráveis a identificação. Ficou estabelecido que há no mercado diversas ferramentas e ambientes de desenvolvimento propícios para a criação de programas de processamento de imagens tornando este tipo de programa acessível para todos os níveis de programadores. O programa apresentado neste trabalho tem objetivo acadêmico, não se mostrando robusto o suficiente para um uso comercial, porém as técnicas utilizadas e os conhecimentos adquiridos podem ser usados para tal fim.

Palavras-chave: identificação de imagem, deep learning, visão computacional, reconhecimento.

Abstract

Image recognition is increasingly present in our routine, and with this advance we have the constant use of biometrics to ensure security of access and identification, thereby technologies such as biometric digital recognition, speech and facial recognition are being widely used, thus making research and development of the topic vital for the advancement of technology. This work demonstrated the development of a facial recognition program using artificial intelligence tools such as machine learning and deep learning in order to correctly identify faces and recognize them, differentiating them from each other. It was observed that the developed program has a good assertiveness, even with the use of images with low resolution and at angles unfavorable to identification. It has been established that there are many tools and development environments on the market that are conducive to the creation of image processing programs making this type of program accessible to all levels of programmers. The program presented in this paper has an academic objective, not being robust enough for a commercial use, but the techniques used and the acquired knowledge can be used for this purpose.

Keywords: image identification, deep learning, computer vision, recognition.

Lista de ilustrações

Figura 1 - Exemplo de conjunto de dados de reconhecimento facial	14
Figura 2 - Equação da taxa de acerto	21
Figura 3 - Função indicadora	21
Figura 4 - Equação para contabilização de erro quadrático médio	21
Figura 5 - Processo de treinamento e avaliação em aprendizagem supervisionada	22
Figura 6 - Representação dos classificadores em cascata utilizado no algoritmo de Viola Jones.....	24
Figura 7 - Imagem em preto e branco de Will Ferrell	28
Figura 8 - Imagem aumentada revelando os pixels da imagem.....	29
Figura 9 - Imagem antes de ser tratada pixel a pixel	29
Figura 10 - Imagem após ser realizado a tratativa pixel a pixel	29
Figura 11 - Imagem após simplificação usando histograma de gradientes orientados.....	30
Figura 12 - Pontos de referencia da Face, segundo o algoritmo face landmark estimation	30
Figura 13 - Resultado obtido após uso do algoritmo Face landmark estimation.....	31
Figura 14 – Exemplo de rosto centralizado 01	32
Figura 15 - Exemplo de rosto centralizado 02	32
Figura 16 - Exemplo de rosto centralizado 03	33
Figura 17 - Rosto centralizado com desconhecidos acertos Exemplo 01	33
Figura 18 - Rosto centralizado com desconhecidos acertos Exemplo 02.....	33
Figura 19 - Rosto centralizado com desconhecidos acertos Exemplo 03	34
Figura 20 - Rosto centralizado com desconhecidos erros Exemplo 01	34
Figura 21 - Rosto centralizado com desconhecidos erros Exemplo 02	34
Figura 22 - Rosto centralizado com desconhecidos erros Exemplo 03	35
Figura 23 - Rostos semi – expostos, perfil ou borrados Exemplo 01	35
Figura 24 - Rostos semi – expostos, perfil ou borrados Exemplo 02	36
Figura 25 - Rostos semi – expostos, perfil ou borrados Exemplo 03	36
Figura 26 - Rostos semi – expostos, perfil ou borrados Exemplo 04	36
Figura 27 - Rostos distorcidos ou encobertos por objetos Exemplo 01.....	37
Figura 28 - Rostos distorcidos ou encobertos por objetos Exemplo 02.....	37
Figura 29 - Rostos distorcidos ou encobertos por objetos Exemplo 03.....	38

Lista de Tabelas

Tabela 1 - Tempos de execução das imagens de teste	38
--	----

Lista de abreviaturas e siglas

API	<i>Aplication Programming Interface</i>
GPU	<i>Graphics Processing Unit</i>
GPL	General Public License
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
MIT	<i>Massachusetts Institute of Technology</i>
RNA	Redes Neurais Artificiais
SO	Sistema Operacional
HOG	<i>Histogram of oriented Gradients</i>

Sumário

1	Introdução	12
<i>1.1</i>	<i>Objetivo</i>	<i>13</i>
<i>1.2</i>	<i>Justificativa</i>	<i>13</i>
<i>1.3</i>	<i>Estrutura do Trabalho</i>	<i>14</i>
<i>1.4</i>	<i>Proposta</i>	<i>14</i>
2	Tecnologias e Ferramentas Utilizadas	16
<i>2.1</i>	<i>Sistema operacional Linux</i>	<i>16</i>
<i>2.2</i>	<i>Inteligência artificial</i>	<i>16</i>
<i>2.3</i>	<i>Redes Neurais</i>	<i>18</i>
<i>2.4</i>	<i>Redes Neurais Convolucionais.....</i>	<i>19</i>
<i>2.5</i>	<i>Aprendizado de máquina</i>	<i>19</i>
<i>2.6</i>	<i>Deep Learning.....</i>	<i>22</i>
<i>2.7</i>	<i>Algoritmo de detecção de faces</i>	<i>22</i>
<i>2.8</i>	<i>Algoritmo Viola Jones</i>	<i>23</i>
<i>2.9</i>	<i>OpenCV.....</i>	<i>24</i>
<i>2.10</i>	<i>Dlib</i>	<i>25</i>
<i>2.11</i>	<i>Redes neurais Convolucionais</i>	<i>26</i>
3	Desenvolvimento	27
<i>3.1</i>	<i>Instalação das ferramentas</i>	<i>27</i>
<i>3.2</i>	<i>Funcionalidade das ferramentas</i>	<i>27</i>
<i>3.3</i>	<i>Fases do reconhecimento Facial.....</i>	<i>28</i>
4	Resultados	32
5	Considerações Finais	39
6	Referências Bibliográfica	40

1 Introdução

Na atual sociedade, o uso de computadores encontra-se amplamente difundido nas diversas camadas da população. Outros fatores que possibilitaram a popularização de dispositivos computacionais: uma maior estabilidade da conexão de *internet*, o crescimento do volume de dados transmitidos, o aumento da velocidade das conexões. Acarretando em um aumento exponencial de aplicações que facilitam a vida de seus usuários, possibilitando assim, que alguns ramos da computação sejam explorados com o intuito de trazer algo que acrescente um ganho, no que se refere a usabilidade e praticidade.

Na vida humana, o reconhecimento de padrões ocorre constantemente a ponto que se torna imperceptível na vida de cada um. Neste exato momento, para ler este texto, seu cérebro está fazendo este reconhecimento para conseguir associar estas imagens como letras e palavras. Até mesmo o processo de conseguir identificar uma pessoa como sendo seu amigo, professor, ou parente realiza um processo de reconhecimento de padrões.

Biometria é a ciência que define a identidade de um indivíduo baseado em seus atributos físicos químicos ou comportamentais (A K JAIN, 2008). A biometria é usada amplamente em sistemas de segurança, tanto que é amplamente usado como um tipo de chave de acesso a aplicações que necessitam a certeza que o indivíduo a estar acessando seja realmente quem tem o acesso, por este motivo o governo estabeleceu a necessidade de existir um registro da digital no documento de identidade, porém a digital não é a única biometria que é única para cada ser humano.

O reconhecimento facial é uma técnica de biométrica que consiste em identificar os padrões existentes na face e em sua composição, dessa forma em um rosto padrão temos uma determinada distância entre os olhos, formato da boca, a linha do maxilar, entre outros. A partir da análise desses padrões de características o ser humano consegue identificar uma grande quantidade de faces, chegando até mesmo ao ponto de identificar faces em objetos inanimados, como nuvens, manchas em paredes e outros ambientes em que não existe uma face. Além disso temos a capacidade de reconhecer uma pessoa familiar mesmo que haja muitos obstáculos na frente. Uma máquina, no entanto, precisa realizar uma grande quantidade de passos para realizar a mesma tarefa.

O reconhecimento de imagens é capacidade de permitir que um determinado *software* possa realizar a identificação de padrões relacionados a uma certa imagem. Através dos anos, algumas técnicas foram desenvolvidas neste âmbito. Uma delas são as Redes Neurais Artificiais (RNA): um método de aprendizado de máquina projetado

para se assemelhar à maneira como o cérebro humano funciona. Com este método, os computadores ensinam a reconhecer os elementos visuais dentro de uma imagem. Ao confiar em bancos de dados grandes e observar padrões emergentes, os computadores podem analisar imagens e formular *tags* e categorias relevantes que fazem sentido. (RAVINDRA. 2017)

Em problemas complexos de reconhecimento de padrões, a abordagem comum utilizada é dividir o problema em dois módulos: a extração de características e o módulo de classificação. Tanto para o primeiro, como para o segundo módulo, já existem várias técnicas desenvolvidas que vão desde estatísticas e matemáticas até técnicas de Inteligência Artificial (IA). (Bishop, 1995)

Neste trabalho, é abordada a construção de um aplicativo de computador, que utiliza um serviço, fornecido pela empresa Intel, que faz o reconhecimento de imagens, onde é feita a identificação dos rostos, assim como também, oferece informações sobre os as pessoas reconhecidas ao usuário da aplicação.

1.1 Objetivo

O objetivo geral deste trabalho é o desenvolver um programa para computadores no sistema computacional Linux, que faça a identificação das faces de pessoas previamente cadastradas com várias fotos do rosto de cada indivíduo, a fim de fazer com que o computador “aprenda” cada face, utilizando uma biblioteca que faz uso de visão computacional e aprendizado de máquina em conjunto com as demais ferramentas descritas no Capítulo 2.

Para isso é necessário atingir alguns objetivos específicos a serem realizados na imagem, são eles a identificação da existência de uma face na imagem selecionada, analisar as características específicas da face encontrada, comparar as características de uma face com as outras já mapeadas para então realizar a previsão utilizando uma nova imagem.

1.2 Justificativa

O intuito originário para a realização deste trabalho parte da oportunidade de facilitar a identificação de astros do cinema, independente do filme que esteja participando. Portanto, a ideia do desenvolvimento de um programa que permita que o usuário descubra em outros vídeos ou fotos, qual ator ou atriz está interpretando os papéis do filme, a aplicação de forma pratica permite que o usuário possa realizar o reconhecimento durante todo o filme desejado. Deste modo, trazendo uma maior praticidade ao dia a dia de seus usuários, já que dúvidas em relação aos atores poderão ser sanadas, pela simples utilização da aplicação. Como exemplo, podemos citar a facilidade que este aplicativo poderá trazer aos novos ingressantes da arte do cinema, que em seus primeiros dias, não sabem exatamente o nome do ator que representa cada

personagem.

1.3 Estrutura do Trabalho

Este documento encontra-se organizado da seguinte forma: No capítulo 1 é apresentado a introdução do trabalho, assim como os motivos que levou ao desenvolvimento do mesmo, explicando os objetivos e justificativas. No Capítulo 2 são apresentadas as tecnologias necessárias para o desenvolvimento do aplicativo. O Capítulo 3 apresenta a proposta de desenvolvimento do aplicativo e suas funções. Testes e resultados objetivos são descritos no capítulo 4. Ao longo do Capítulo 5 temos as considerações finais sobre este trabalho. E no capítulo 6 temos as referências bibliográficas.

1.4 Proposta

A proposta do presente trabalho é o desenvolvimento de um aplicativo na plataforma Linux, que possui como funcionalidade principal, a disponibilização de informações sobre quem é o ator ou pessoa que está nas fotos ou vídeos desejados pelo usuário, apenas ao colocar o caminho do arquivo de vídeo ou foto desejado. Demonstrando a localização do rosto de cada pessoa e escrevendo o nome dela debaixo de cada marcação.

Com o intuito do programa informar o nome de cada pessoa encontrada no arquivo desejado, o usuário terá que carregar anteriormente as fotos dessa pessoa para que o programa computacional aprenda a reconhecer o dono da face.

Para a presente aplicação foi escolhido o filme “Jurassic Park (1993)” e os seguintes atores do filme: Alan Grant, paleontologista (22 imagens); Claire Dearing, gerente de operações do parque (53 imagens); Ellie Sattler, paleobotânica (31 imagens); Ian Malcolm, matemático (41 imagens); John Hammond, empresário / proprietário do Jurassic Park (36 imagens); Owen Grady, pesquisador de dinossauro (35 imagens).

Abaixo temos o exemplo utilizado, um conjunto de dados de reconhecimento de rosto criado programaticamente com o Python e a API de pesquisa de imagens do Bing. São mostrados seis dos personagens da série de filmes do Jurassic Park.

Figura 1 - Exemplo de conjunto de dados de reconhecimento facial

Alan Grant



Claire Dearing



Ellie Sattler



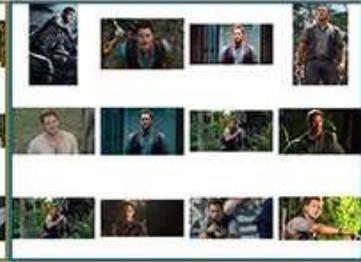
Ian Malcolm



John Hammond



Owen Grady



2 Tecnologias e Ferramentas Utilizadas

Neste Capítulo, serão descritas as ferramentas utilizadas na construção da aplicação, assim como a motivação para suas utilizações no projeto.

Inicia-se com uma descrição do sistema operacional Linux, utilizado na codificação do programa, os conceitos de inteligência artificial e *machine learning* e as tecnologias necessárias para a criação do programa, assim como os algoritmos utilizados no reconhecimento de faces e de padrões.

2.1 Sistema operacional Linux

O Linux foi criado por Linus Torvalds em 1991 e desde sua concepção é um sistema operacional de código aberto, porém para isso ser possível é necessário ao Linux possuir uma licença GnuPublicLicense (GPL). Com esta licença é possível que o sistema operacional seja distribuído por qualquer um, por esse motivo temos grandes empresas como Intel, IBM, HP, Hitachi, entre varias outras que ajudam o projeto Linux o financiando.

Linux é um sistema de código aberto, por esse motivo é comum que diversas plataformas o utilizem, não só para plataformas de computador, mas também de smartphones e outros tipos de hardware como televisores, centrais de carros, câmeras fotográficas, entre outros (JEONG, 2012).

O sistema operacional Linux é um sistema operacional monolítico, e possui gerenciamento de memória, suporte a vários sistemas de arquivos e suporte a uma gama enorme de dispositivos de hardware. O código fonte do kernel Linux segue uma estrutura de pasta definida pelo seu conteúdo (RUSLING, 1999). A maior parte do código fonte se concentra no diretório drivers, o qual contém o código fonte de todos os drivers de dispositivos de todo o sistema (desde drivers para dispositivos de rádio, vídeo, teclado, controles de vídeo games, e todo o tipo de dispositivos com suporte ao Linux).

2.2 Inteligência artificial

A inteligência artificial está em muitos ramos de estudo, e seu amplo estudo e desenvolvimento fez com que defini-lo seja algo complicado, dessa forma é mais simples descrever as quatro linhas de pensamento que surgiram no decorrer da

evolução da inteligência artificial.

Segundo Rich e Knight (1993), inteligência artificial (IA) é o estudo de como fazer os computadores realizarem tarefas que, até o presente, os seres humanos realizam melhor. Esta definição entende que a máquina precisa agir como um ser humano, porém não consegue incluir problemas que não podem ainda ser bem solucionados nem pelos computadores nem pelas pessoas, mas cria uma noção do que vem a ser inteligência artificial.

De acordo com Rabuske (1995, p. 21), inteligência artificial é o resultado da aplicação de técnicas e recursos especialmente de natureza não numérica, tornando possível a solução de problemas que exigiriam do humano certo grau de raciocínio e de perícia. Geralmente a solução destes problemas é muito mais difícil com recursos tipicamente numéricos. Dessa forma a máquina precisa agir com racionalidade para solucionar os problemas propostos.

John Haugeland (Haugeland, 1985), definiu inteligência artificial como a capacidade de uma máquina em pensar, dessa forma estabelecendo que as máquinas precisam evoluir ao ponto de possuir mentalidade própria, no sentido total e literal, sendo assim as máquinas devem pensar como seres humanos.

Segundo Charniak (CHARNIAK; MCDERMOTT, 1985), IA é o estudo das faculdades mentais através do uso de modelos computacionais, sendo assim definindo que as máquinas dotadas de IA precisam pensar de forma racional.

Com os conceitos em mente entendemos que duas linhas de pensamento (Charniak e Haugeland), nos levam a encarar a IA como a forma que os computadores pensam e raciocinam enquanto as outras duas (Rabuske e Rich e Knight), define a IA como a forma comportamental que a máquina age.

Com as duas vertentes definidas podemos falar sobre Alan Turing e o Teste de Turing, que buscava definir uma forma de quantificar se uma máquina tinha a capacidade de se comportar como um ser humano, ao ponto de passar por uma pessoa. O teste consistia em um interrogador (humano), trocar mensagens, uma conversa com duas entidades da qual ele não pode ver, sendo assim as mensagens precisam ser feitas por um meio estático como via teclado. As duas entidades são compostas por um ser humano e uma máquina dotada de IA e os dois irão tentar convencer o interrogador de que são humanos. A máquina pode ser considerada realmente dotada de inteligência artificial no momento em que o interrogador não conseguiu afirmar qual dos dois é uma máquina.

A partir da definição do Teste de Turing foram estabelecidos alguns requisitos

mínimos que uma máquina precisa ter para conseguir passar pelo teste, entre eles ter a capacidade de falar um idioma, de forma que a comunicação seja viável; A capacidade de armazenar conhecimento, seja antes ou durante o teste; Raciocínio automático, a fim de associar as perguntas feitas ao conhecimento adquirido para responder da forma correta as perguntas; e a capacidade da máquina de aprender com as experiências anteriores e se adaptar a partir delas.

Projetando esse objetivo ao limite, espera-se que, no futuro, seja possível interagir com um programa de computador e não saber se o interlocutor é de fato a máquina ou uma outra pessoa.

Segundo Santos (2002, p. 15), é exatamente isso que o jogador espera do comportamento dos personagens do jogo: que seus atos, suas reações e seu diálogo sejam o mais parecido possível com os de uma pessoa real. E, além disso, devido ao impressionante desenvolvimento da computação gráfica para jogos, o jogador passou a ter expectativas de que o mesmo grau de realismo seja observado no comportamento dos personagens do jogo.

2.3 Redes Neurais

Uma rede de computação neural é um modelo matemático que consiste em várias unidades simples, semelhantes aos neurônios, denominados elementos de processamento. Estes elementos são organizados em camadas, onde cada elemento tem conexões ponderadas com outros elementos, sendo que estas conexões podem tomar diferentes configurações, dependendo da aplicação particular desejada (LOESCH; SARI, 1996). Segundo Luger (2004, p 391), para construir uma rede neural, por exemplo, o projetista deve criar um esquema para codificar padrões do mundo em quantidades numéricas da rede. A escolha de um esquema de codificação desempenha um papel crucial no sucesso ou fracasso da rede em aprender.

Redes neurais podem possuir diferentes arquiteturas e diferentes métodos de aprendizado. Segundo Luger (2004, p. 395), o modelo Perceptron, por exemplo, utiliza uma forma simples de aprendizado supervisionado. Durante o processo de treinamento de uma rede neural várias entradas são informadas. Durante esta etapa é analisado se para uma determinada entrada a saída desejada é obtida. Caso o valor de saída não for satisfatório os pesos das conexões são reajustados até que a rede passe a mapear corretamente a saída para uma entrada informada ou um erro aceitável seja alcançado. Este procedimento tem por objetivo reduzir o erro médio sobre todo o conjunto de treinamento. Após o período de treinamento uma rede

neural pode atingir uma configuração de pesos que é capaz de classificar as entradas conforme o resultado de saída desejado. A fase de utilização é a próxima etapa, nela os pesos não são mais ajustados. Nesta fase uma entrada é informada, analisada e o resultado é retornado (LUGER, 2004, p.395).

2.4 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (Convolutional Neural Network - CNN) são arquiteturas de aprendizado profundo que extrai as características de cada conjunto subdividindo os dados. Com o intuito de reduzir o número de parâmetros que deverão ser ajustados pela rede posteriormente, melhorando assim, o processo de treinamento. Um aspecto importante desta arquitetura que a difere das demais é que ela está relacionada com sua invariância a escala, a translação e outras transformações, ou seja, ela consegue reconhecer padrões de forma mais robusta e automática.

As CNNs são MLNN muito utilizadas em tarefas com estruturas em grades, como por exemplo, o processamento de fala e entendimento da linguagem natural (1D, convoluções temporais), segmentação e classificação de imagens (2D, convolução espacial), e análise de vídeos (3D, convolução) (SIMONOVSKY; KOMODAKIS, 2017) (LECUN; BENGIO; HINTON, 2015).

Apesar da CNN ser mais rápida e eficaz que a utilização da Dlib, ela requer GPU o suficiente para realizar os cálculos e renderizar as imagens. Visando otimizar o tempo, foi decidido utilizar Dlib no programa em questão.

2.5 Aprendizado de máquina

O aprendizado de máquina é um subcampo da inteligência artificial, um campo que surgiu dentro da ciência da computação, e aborda a questão sobre como tornar as máquinas aptas a aprender. No contexto em que se insere, o aprendizado se refere à inferência indutiva (RÄTSCH, 2004).

O objetivo principal do aprendizado de máquina é generalizar além dos exemplos existentes no conjunto de treinamento, pois independentemente da quantidade de dados existentes é muito improvável que, durante os testes, exatamente os mesmos exemplos apareçam.

Ainda não é possível fazer computadores aprenderem tão bem quanto as pessoas, porém algoritmos criados são eficientes para várias tarefas de aprendizado, e os estudos teóricos sobre aprendizado estão permitindo que novas técnicas sejam desenvolvidas. (MITCHELL, 2006)

Um dos aspectos que tornam esse um tema de importância é a dimensão dos

problemas a serem resolvidos. A intuição humana é treinada em um universo tridimensional, logo existe uma dificuldade natural de se resolver problemas de dimensões maiores sem a utilização de ferramentas adequadas. (DOMINGOS, 2012) Além disso, a capacidade de armazenamento dos computadores e a quantidade de dados disponíveis cresce vertiginosamente e há muito o ser humano não consegue processar esse grande volume sem o auxílio de ferramentas de software.

Apesar da grande dimensão, inerente a certos problemas, ser uma das principais motivações de utilizar algoritmos de aprendizado de máquina, ela traz uma dificuldade para a avaliação dos resultados obtidos com os modelos gerados por esses algoritmos, já que muitas vezes a intuição não é suficiente avaliar certos aspectos. Para tanto, técnicas para avaliação dos algoritmos e modelos gerados são muito estudadas e necessárias para permitir entender o grau de confiança que se pode empregar sobre os resultados obtidos.

O aprendizado de máquina pode ser dividido em dois importantes subgrupos de algoritmos: os que compõe a aprendizagem não-supervisionada e a aprendizagem supervisionada.

2.5.1 Aprendizagem não-supervisionada

A aprendizagem não-supervisionada, por vezes chamada modelagem descritiva, não utiliza informações das variáveis de saída. Os dados de entradas são analisados e agrupados conforme a proximidade dos seus valores. Para cada grupo, ou cluster, um rótulo é utilizado, permitindo indicar a qual grupo cada registro pertence. (DUDA, 2001)

2.5.2 Aprendizagem supervisionada

Como o próprio nome propõe, na aprendizagem supervisionada é necessário a presença de um supervisor que é dado pelo registro dos valores das variáveis de saída, ou variáveis-objetivo. Essas variáveis precisam ser escolhidas de modo que represente a resposta para algum problema a ser resolvido (SATHYA, 2013). Dessa forma obtém-se um modelo que descreve um conjunto de dados e a partir dele seja possível prever o comportamento de saída para novas entradas.

Os principais algoritmos de aprendizagem supervisionada criam modelos que permitem a classificação, para variáveis categóricas, e a regressão, para variáveis contínuas. A classificação consiste em atribuir um rótulo para a saída a partir de determinada entrada. A regressão retorna um valor que pertence a um espectro contínuo de valores. (MOHRI, 2012)

Para avaliar o modelo obtido, normalmente se separa uma parte dos exemplos existentes para realizar testes, de preferência não os utilizando durante o processo de treinamento para evitar o sobre ajuste, ou *overfitting*. Esse fenômeno ocorre quando o modelo descreve muito bem o conjunto utilizado para treinamento, mas tem baixa capacidade de prever novas entradas. É importante ressaltar que os dados inerentemente apresentam desvios causados por fatores aleatórios, e os algoritmos devem minimizar a influência desses desvios na geração dos modelos.

Para classificadores, a técnica mais utilizada para a avaliação consiste em comparar os valores de saída obtidos com o modelo treinado, utilizando os exemplos de teste como entrada, com os valores de saída existentes nesses exemplos. A taxa de acerto, ou *accuracy*, é dada por:

Figura 2 - Equação da taxa de acerto

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \mathbb{1}(\hat{y}_i = y_i)$$

onde y é a saída existente nos exemplos de teste, \hat{y} os valores obtidos com o modelo treinado, $n_{samples}$ o número de exemplos de teste e $\mathbb{1}(x \in A)$ é a função indicadora, dada por:

Figura 3 - Função indicadora

$$\mathbb{1}(x \in A) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$$

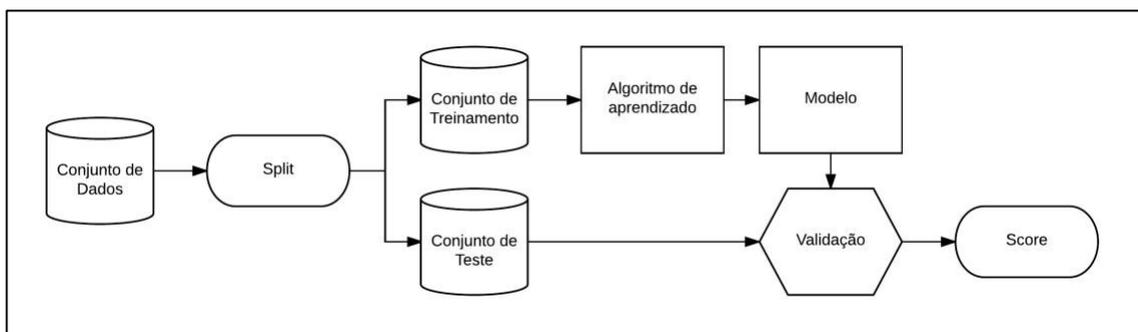
Já para modelos de regressão, o erro pode ser contabilizado utilizando o erro quadrático médio, dado por:

Figura 4 - Equação para contabilização de erro quadrático médio

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (\hat{y}_i - y_i)^2$$

Um diagrama que representa o processo de treinamento e avaliação é apresentado na Figura 5 a seguir.

Figura 5 - Processo de treinamento e avaliação em aprendizagem supervisionada



2.6 Deep Learning

De acordo com Deng (2014), a arquitetura de *deep learning* (aprendizado profundo) consiste em conjunto de técnicas de aprendizado de máquina, focados na utilização de diversas camadas de processamento de informação, sendo aplicado com sucesso em diversas áreas, tais como visão computacional e reconhecimento de objetos. Estas arquiteturas utilizam redes com diversas camadas intermediárias, sendo que através delas, segundo Wu et al. (2015), tornou-se então possível reconhecer objetos visuais sem a necessidade da extração prévia de características.

Sendo uma subcategoria do aprendizado de máquina, o aprendizado profundo se caracteriza por ser capaz de aprender com suas experiências, por causa disso é necessária uma baixa intervenção humana para a mesma adquirir conhecimento, e aprender. A hierarquia de conceitos faz com que problemas muito complexos possam ser divididos em problemas mais simples, facilitando assim o desenvolvimento e entendimento do programa. Assim, se tivéssemos que ilustrar esse conceito, um grafo poderia exemplificá-lo bem, devido a existência de várias camadas, onde a cada nível está ligado ao nível anterior, aumentando a sua complexidade a cada novo nível. Para casos onde há um grande número de elementos a serem estudados envolvendo aprendizado de funções e de conceitos, o aprendizado profundo se sai melhor que o aprendizado de máquina, devido a sua capacidade de entender relações mais complexas (YOSHUA, 2016).

A capacidade do aprendizado profundo de utilizar funções não lineares juntamente do entendimento de conceitos mais complexos pelo uso de representação hierárquica através de uma base de dados não rotulados permite ao algoritmo encontrar padrões (LI DENG, 2014), onde seria difícil de entendimento para os seres humanos, como por exemplo associar quais elementos de um texto pode ser usados para determinar um Fake News ou não.

2.7 Algoritmo de detecção de faces

Reconhecimento facial é uma tarefa simples para os humanos que até mesmo

um bebê com poucos dias de vida já consegue 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.

A detecção de faces é uma das principais tarefas executadas durante uma série de outras aplicações como rastreamento de faces, identificação de faces e remoção de olho vermelho (YANG; KRIEGMAN; AHUJA, 2002), pois antes de qualquer um destes procedimentos citados é necessário delimitar a área da imagem na qual encontra-se a face para então dar sequência em um processamento para identificação, por exemplo. Dessa forma, a utilização de um algoritmo eficiente para detectar a face antes de realizar a identificação facial, por exemplo, implicará diretamente na eficiência da aplicação de identificação, já que um erro na primeira etapa culmina com um erro no reconhecimento.

A princípio não parece muito complexo, porém uma série de desafios estão associados a este processo (YANG; KRIEGMAN; AHUJA, 2002). Dentre os fatores que podem dificultar o procedimento, tem-se:

- A pose da pessoa na foto (frontal, perfil etc.)
- Presença ou não de barba, bigode, óculos.
- Expressões faciais.
- Orientação da câmera.
- Condições do ambiente (luz, brilho etc.)

Existem diversos algoritmos utilizados para a detecção de objetos, dentro os quais optou-se por utilizar o algoritmo HOG (Histogram of Oriented Gradients) muito utilizado para detectar faces.

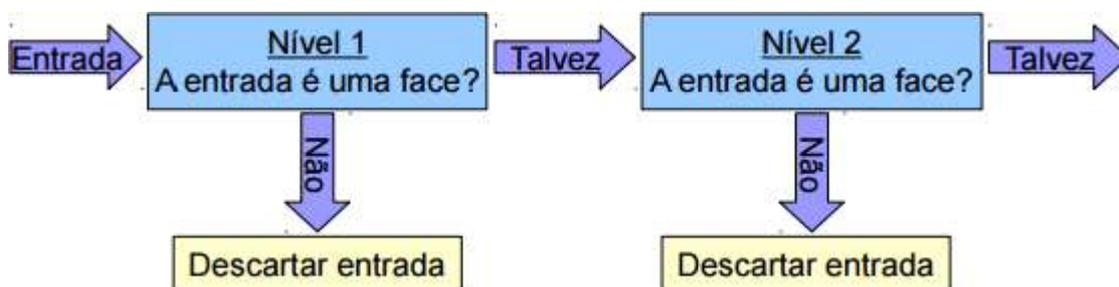
2.8 Algoritmo Viola Jones

Nos últimos anos, obteve-se muito progresso em técnicas (DEVIS; MANE; AJAYKUMAR, 2015) para detecção facial. Dentre as pesquisas realizadas, o detector de objetos Viola-Jones (VIOLA; JONES, 2001) conseguiu grande destaque devido a sua eficiência em detectar faces com elevada taxa de acertos, baixa taxa de falso positivo e principalmente devido ao seu baixo custo computacional (ARAÚJO, 2010).

Subdividido em três partes, o algoritmo de Viola-Jones primeiramente

utiliza filtros de Haar para representar a imagem em um espaço característico. Isto é realizado utilizando-se a “Imagem Integral” (VIOLA; JONES, 2001) a qual é responsável por aumentar o ganho computacional pois as operações são realizadas sobre retângulos ao invés de operar sobre pixels. Em um segundo passo, utiliza-se um classificador baseado método AdaBoosting (VIOLA; JONES, 2001) responsável por selecionar as características mais relevantes extraídas no passo anterior. Para obter um melhor desempenho utiliza-se classificadores em cascata. A Figura 2 ilustra o funcionamento dos classificadores em cascata.

Figura 6 - Representação dos classificadores em cascata utilizado no algoritmo de Viola Jones



Para realizar a detecção utiliza-se as características selecionadas no passo anterior, as quais são escalonadas em diversos tamanhos já que não se sabe o tamanho nem a posição da face na imagem alvo, e então são aplicadas em todas sub janelas da imagem para verificar se a característica está presente ou não na imagem. Há uma série de classificadores já definidos e salvos em arquivos XML que podem ser utilizados diretamente por ferramentas como OpenCV.

2.9 OpenCV

OpenCV é uma biblioteca de visão computacional de código aberto, projetada para eficiência computacional com grande foco em aplicações de tempo real (BRADSKI; KAEHLER, 2008). A biblioteca OpenCV foi desenvolvida pela Intel e possui mais de 500 funções que abrangem diversas áreas de visão computacional (MARENGONI; STRINGHINI, 2009). A OpenCV possui interfaces para as linguagens C++, Python e Java, com suporte aos sistemas operacionais Windows, Linux, Mac OS, iOS e Android (OPENCV, 2018).

De acordo com Bradski e Kaehler (2008), visão computacional é a

transformação de dados de uma imagem estática ou vídeo em uma decisão ou uma nova representação. Qualquer uma das transformações visa atingir algum objetivo em particular. Um exemplo de decisão seria verificar a presença de um carro em uma determinada imagem ou identificar a quantidade de rostos presentes em uma imagem. Um exemplo de nova representação seria a estabilização de um vídeo ou transformar uma imagem colorida em uma imagem monocromática.

Segundo Bradski e Kaehler (2008), um dos objetivos da biblioteca OpenCV é prover uma infraestrutura de visão computacional de fácil uso, para ajudar pessoas a desenvolverem aplicações sofisticadas de visão computacional rapidamente. Utilizada em diferentes regiões em todo o mundo, a biblioteca OpenCV possui uma comunidade superior a 47 mil usuários e seu número de downloads é estimado em mais de 14 milhões (OPENCV, 2018). Desde seu lançamento em 1999, a biblioteca OpenCV tem sido utilizada em diversas aplicações, produtos e pesquisas. Essas aplicações incluem monitoramento automático de sistemas de segurança, redução de ruído de imagens na área da medicina, calibração de câmeras, aplicações militares e muitas outras (BRADSKI; KAEHLER, 2008). Essa biblioteca irá carregar, converter e mostrar a imagem processada, por ter um desempenho menor no reconhecimento facial comparado ao Dlib.

2.10 Dlib

A Dlib é uma biblioteca de *machine learning*, escrita em C++, que implementa 14 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. 15 Foi escolhida por ser mais precisa e eficiente que a biblioteca OpenCV no 16 reconhecimento facial ao utilizar a técnica HOG (*Histogram of Oriented Gradients*) que será especificada no desenvolvimento do programa, em conjunto com um classificador treinado para detectar faces (especificamente, *support vectormachine*), e utilizar um modelo pré-treinado de uma Rede Neural Convolutiva na hora de gerar o vetor descritor da face (a partir do qual o reconhecimento propriamente dito pode ser alcançado através do simples cálculo de distância entre vetores descritores). Para um descritor de 128 dimensões, alcançou uma precisão de 99.38% (KING,2017) no *dataset Labeled Faces in the Wild* (HUANG; RAMESH; BERG; MILLER, 2007).

2.11 Redes neurais Convolucionais

As Redes Neurais Convolucionais (*Convolutional Neural Network - CNN*) são arquiteturas de aprendizado profundo que extrai as características de cada conjunto subdividindo os dados. Com o intuito de reduzir o número de parâmetros que deverão ser ajustados pela rede posteriormente, melhorando assim, o processo de treinamento. Um aspecto importante desta arquitetura que a difere das demais é que ela está relacionada com sua invariância a escala, a translação e outras transformações, ou seja, ela consegue reconhecer padrões de forma mais robusta e automática.

As redes neurais convolucionais são redes neurais multicamadas muito utilizadas em tarefas com estruturas em grades, como por exemplo, o processamento de fala e entendimento da linguagem natural (1D, convoluções temporais), segmentação e classificação de imagens (2D, convolução espacial), e análise de vídeos (3D, convolução) (SIMONOVSKY; KOMODAKIS, 2017) (LECUN; BENGIO; HINTON, 2015).

Apesar da rede neural convolucional ser mais rápida e eficaz que a utilização da Dlib, ela requer *GPU* o suficiente para realizar os cálculos e renderizar as imagens. Visando otimizar o tempo, foi decidido utilizar Dlib no programa em questão.

3 Desenvolvimento

Neste capítulo serão descritos, detalhadamente, todos os procedimentos utilizados na construção da aplicação, iniciando pela instalação dos programas necessários para iniciar a construção do aplicativo, seus métodos e interfaces e a utilização do serviço *opencv* e *dlib*, que realizou todo o processo oneroso de identificação das imagens, possibilitado através da utilização de sua biblioteca.

3.1 Instalação das ferramentas

O sistema operacional Ubuntu, oferece uma forma rápida, simples e fácil de instalar os pacotes necessários para a realização do programa. Utilizando o comando `-pip`, é possível baixar e instalar os pacotes desejados diretamente da fonte onde eles se encontram.

Foram utilizados os seguintes comandos:

- `pip install opencv-contrib-python`
- `pip install dlib`
- `pip install face_recognition`
- `pip install imutils`

3.2 Funcionalidade das ferramentas

O comando *opencv-contrib-python* instala no sistema operacional os recursos para a programação na linguagem python, assim como a biblioteca do *opencv* para poder realizar a manipulação das imagens reais no disco, mas é organizada por módulos e tipos de arquivo para simplificar o projeto.

O *Dlib* é um kit de ferramentas C++ moderno que contém algoritmos e ferramentas de aprendizado de máquina para criar softwares complexos em C++ para resolver problemas do mundo real. Ele é usado tanto na indústria quanto na academia em uma ampla gama de domínios, incluindo robótica, dispositivos incorporados, telefones celulares e grandes ambientes de computação de alto desempenho.

O “*face_recognition*” é a biblioteca utilizada para reconhecer e manipular faces do Python ou da linha de comando. Construído usando o reconhecimento de face de última geração do *dlib*, construído com *deep learning*. O modelo tem uma precisão de 99,38% no benchmark *Labeled Faces in the Wild*.

E o “*imutils*”, é uma biblioteca com uma série de funções de conveniência para tornar as funções básicas de processamento de imagens, como tradução, rotação,

redimensionamento, esqueletização e exibição de imagens do Matplotlib mais fáceis com o OpenCV e o Python 2.7 e o Python 3.

3.3 Fases do reconhecimento Facial

Para realizar o correto funcionamento do programa foi necessário seguir etapas para o tratamento da imagem.

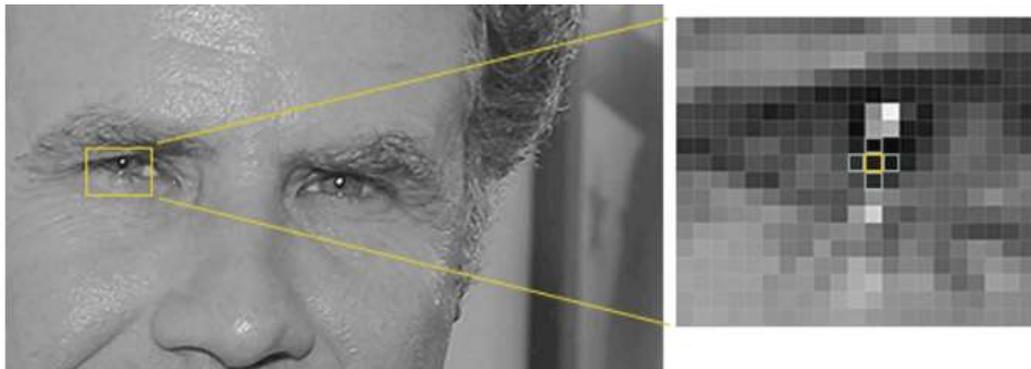
Para iniciar é necessário identificar se há uma face na imagem, para isso começamos transformando a imagem em uma imagem em preto e branco pois para realizar a identificação não precisamos da informação de cores, além de reduzirmos o “peso” da imagem.

Figura 7 - Imagem em preto e branco de Will Ferrell



O próximo passo é analisar a imagem pixel a pixel de forma que seja possível definir o quão escuro os pixels que cercam o pixel a estar sendo analisado são, e assim desenhamos uma seta no sentido em que o conjunto de pixel é mais escuro.

Figura 8 - Imagem aumentada revelando os pixels da imagem

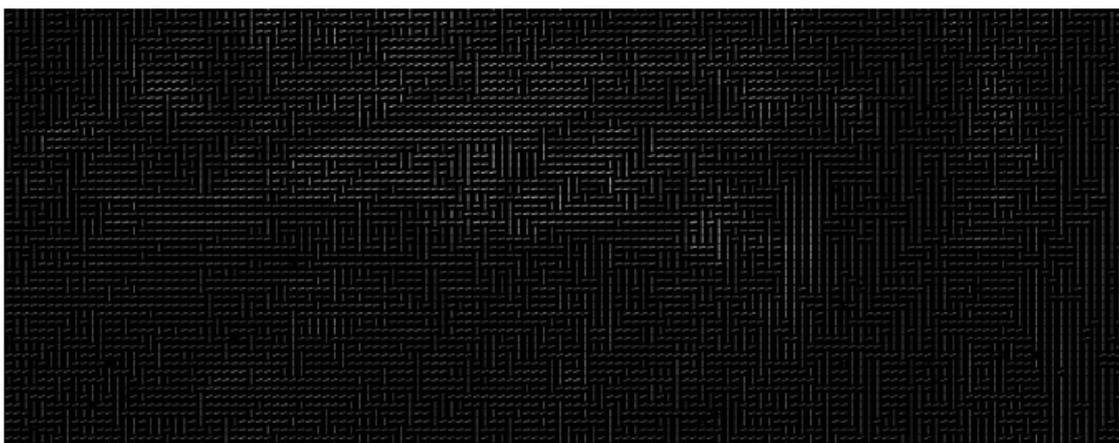


A princípio a realização desta etapa não faz muito sentido, porém ao ser realizado para todos os pixels da imagem chegamos a uma imagem próxima da figura 10 abaixo.

Figura 9 - Imagem antes de ser tratada pixel a pixel

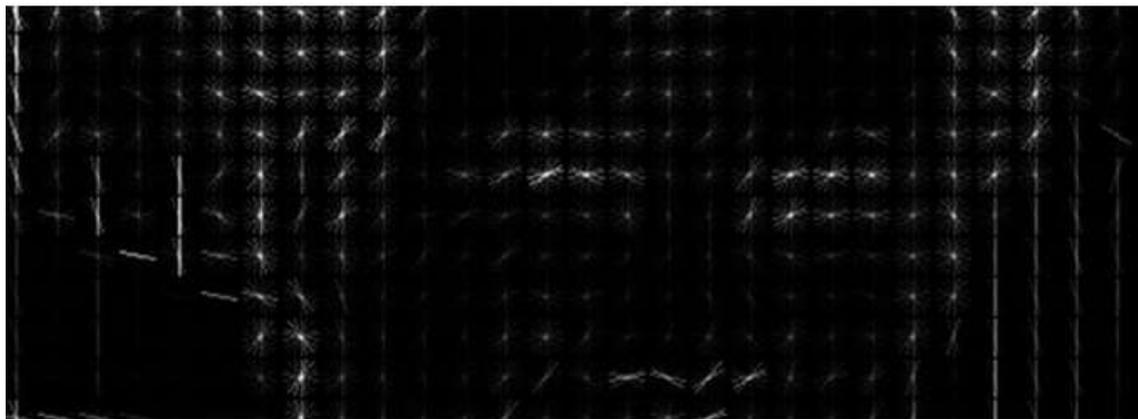


Figura 10 - Imagem após ser realizado a tratativa pixel a pixel



Feito isto podemos trabalhar a imagem de forma que apenas os traços mais básicos são considerados, sendo assim a imagem é dividida em conjunto de 16x16 pixels onde são contadas as setas que apontam numa determinada direção de forma que possamos usar apenas aquelas que aparecem com uma maior frequência em cada conjunto de 16x16 pixel. Com isso teremos uma imagem parecida com a figura 11.

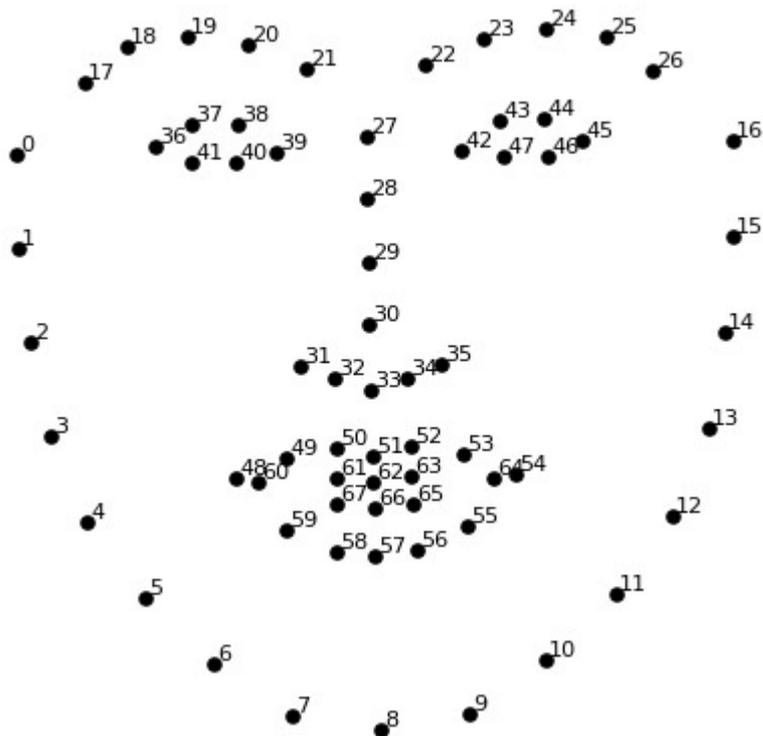
Figura 11 - Imagem após simplificação usando histograma de gradientes orientados



Feito essa tratativa precisamos solucionar o problema de que as faces podem não estar bem posicionadas na imagem ou as características principais da face podem não estar na imagem, sendo assim precisamos mapear as características principais e validar a relação entre essas características.

Para isso foi utilizado um algoritmo chamado *face landmark estimation*, criado em 2014 por Vahid Kazemi and Josephine Sullivan. Basicamente o algoritmo mapeia 68 pontos principais da face, são eles a linha do queixo, a parte exterior dos olhos, a parte interna da curva da sobrancelha, entre outros, na figura 12 temos um exemplo.

Figura 12 - Pontos de referencia da Face, segundo o algoritmo *face landmark estimation*



Considerando nossa a figura 7 após utilizar o algoritmo citado acima teremos

algo parecido com a figura 13.

Figura 13 - Resultado obtido após uso do algoritmo Face landmark estimation



Realizado a identificação das faces é possível realizar a diferenciação das faces, de modo que ao ser comparada com outra face seja possível diferencia-la, para isso precisamos analisar as características de cada face e criar uma correlação entre as imagens que sejam da mesma pessoa.

Para realizar a identificação foram usadas 128 medições para cada imagem, de forma que ao comparar duas imagens da mesma pessoa os dados gerados pelas 128 medições são suficientemente próximos para definir quem é a pessoa da foto, uma vez que aquela face já tenha sido previamente analisada. Dessa forma o treinamento consiste em inserir uma imagem de uma pessoa, onde é retirado as 128 medições, depois é inserida outra foto da mesma pessoa e feito o mesmo processo e por fim uma terceira foto, dessa vez de outra pessoa e realizado o mesmo processo. Analisando os dados é possível definir que as duas primeiras medições são mais próximas entre si que a terceira medição realizada.

4 Resultados

Após a instalação das ferramentas necessárias, a criação das imagens usadas de modelo para que o reconhecimento, a finalização do projeto e o funcionamento foi obtido os seguintes resultados:

Para imagens claras e centralizadas dos rostos foi obtido bons resultados, com o programa conseguindo identificar os rostos de cada personagem, observem as figuras 14, 15 e 16.

Figura 14 – Exemplo de rosto centralizado 01

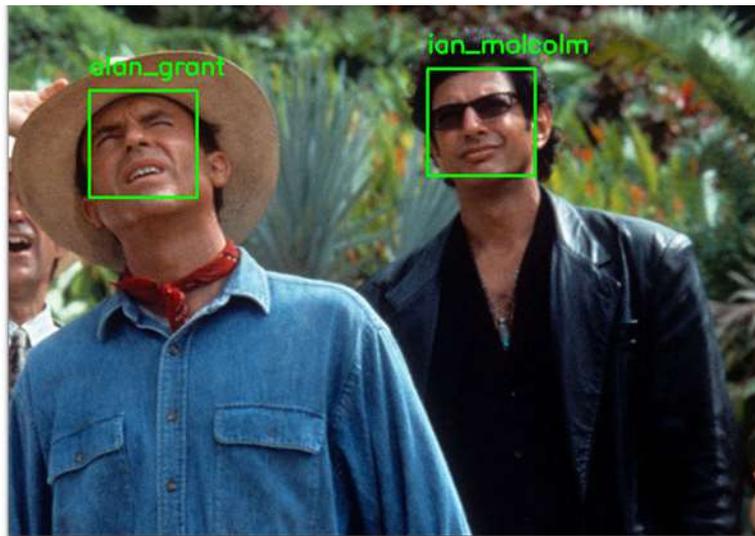


Figura 15 - Exemplo de rosto centralizado 02

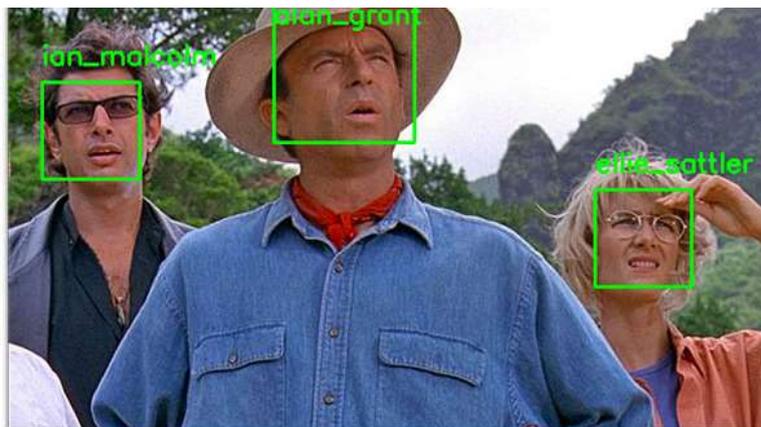
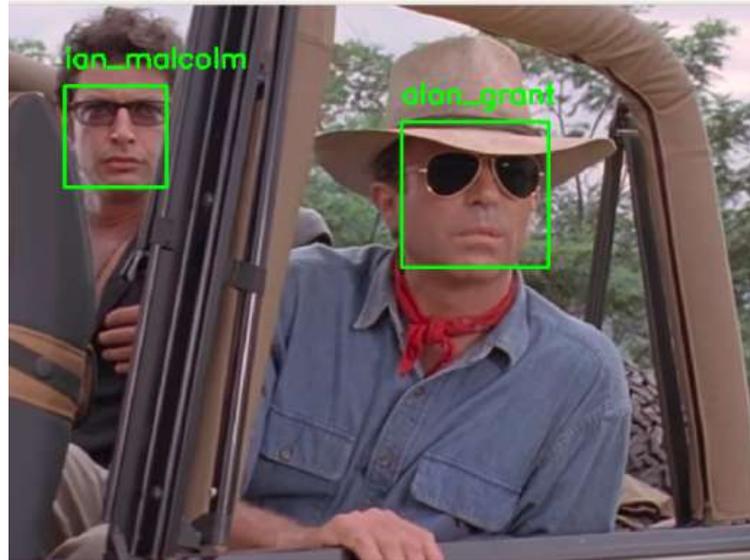


Figura 16 - Exemplo de rosto centralizado 03



Foram realizados também realizados testes com rostos centralizados, porém onde havia personagens que não foram previamente reconhecidos, segue os exemplos das figuras 17, 18 e 19.

Figura 17 - Rosto centralizado com desconhecidos acertos Exemplo 01

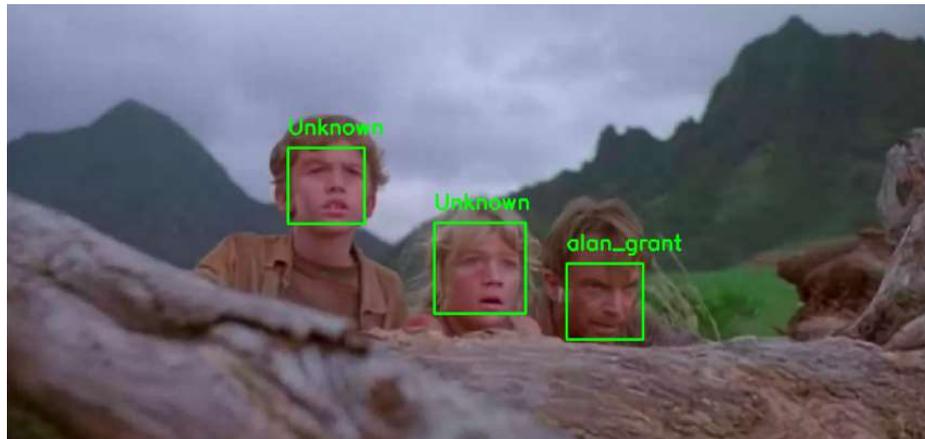


Figura 18 - Rosto centralizado com desconhecidos acertos Exemplo 02

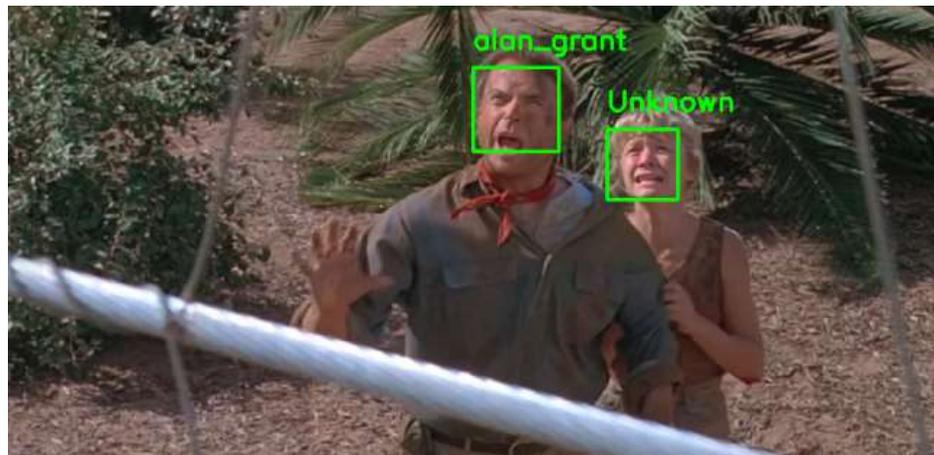
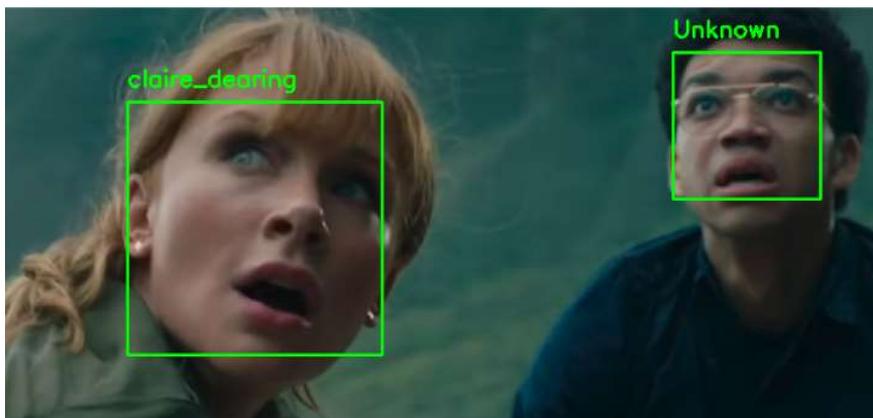


Figura 19 - Rosto centralizado com desconhecidos acertos Exemplo 03



Porém não foram todas as situações em que ao adicionar um rosto desconhecido o programa conseguiu identificar corretamente, em algumas imagens onde o rosto estava um pouco mais embaçado, ou a semelhança entre os personagens desconhecidos e os conhecidos eram grande o programa falhou em identificar corretamente como desconhecido, segue os exemplos figura 20, 21 e 22.

Figura 20 - Rosto centralizado com desconhecidos erros Exemplo 01

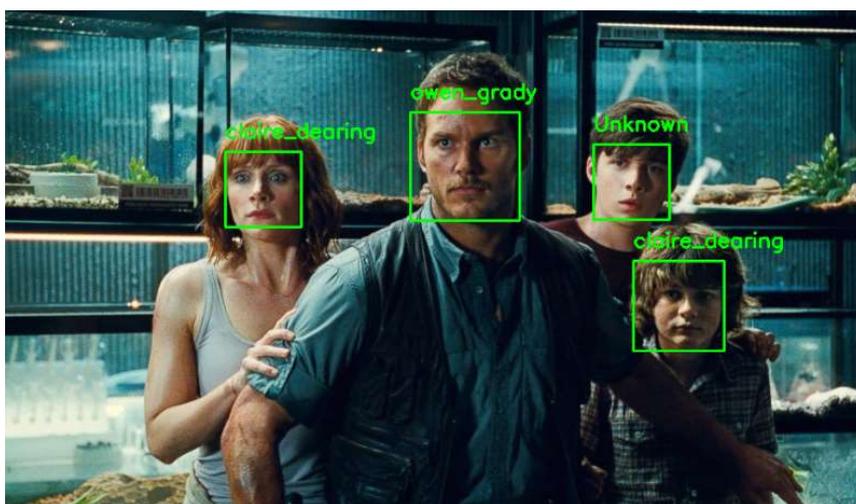
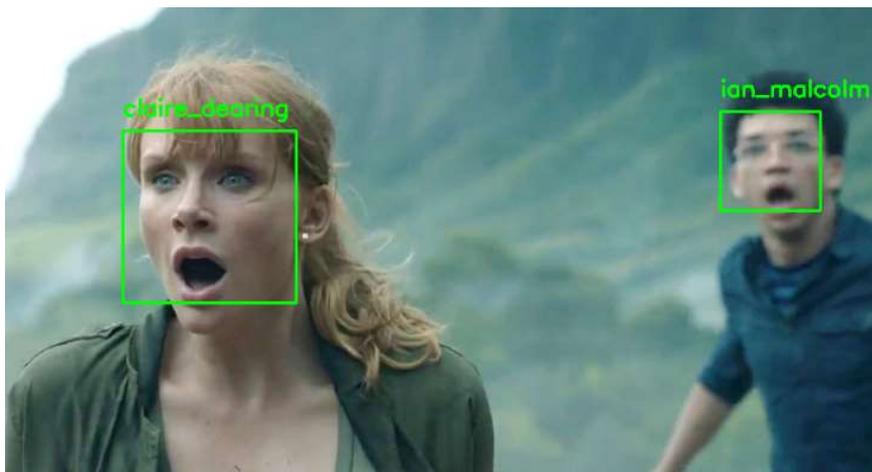


Figura 21 - Rosto centralizado com desconhecidos erros Exemplo 02



Figura 22 - Rosto centralizado com desconhecidos erros Exemplo 03



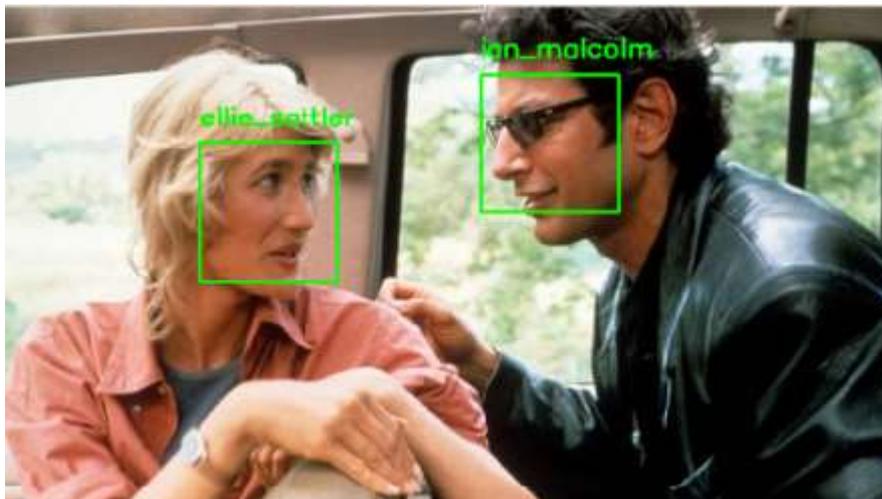
O programa foi eficiente em reconhecer os rostos em imagens muito embaçadas, em perfil ou quando o rosto dos personagens estava semi-expostos, segue como exemplo a Figura 23 onde apesar da imagem estar escura e o rosto de Owen Grady estar borrado, ainda assim o programa foi eficiente em identifica-lo.

Figura 23 - Rostos semi – expostos, perfil ou borrados Exemplo 01



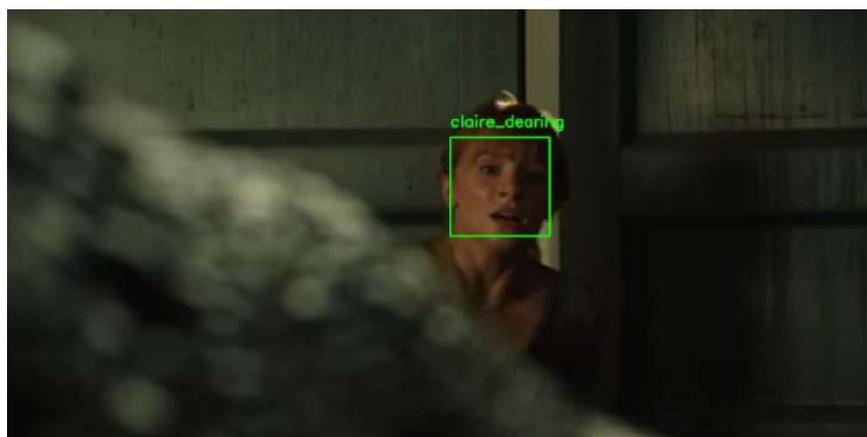
Na Figura 24 temos os rostos de Ellie Bottler e Ian Malcolm em perfil, dificultando assim a identificação de todas as características da face, porém o programa novamente foi eficiente em reconhece-los

Figura 24 - Rostos semi – expostos, perfil ou borrados Exemplo 02



Na figura 25 o rosto de Claire Dearing está com uma grande sombra, tornando o mesmo muito escuro no lado esquerdo da personagem, ainda assim o programa conseguiu identifica-la corretamente.

Figura 25 - Rostos semi – expostos, perfil ou borrados Exemplo 03



O ultimo exemplo é a Figura 26 onde o rosto de Alan Grant foi corretamente reconhecido em uma imagem muito borrada e com pouca resolução

Figura 26 - Rostos semi – expostos, perfil ou borrados Exemplo 04



Nos testes realizados houve imagens que estavam centralizadas, porém haviam objetos alterando a forma do rosto. Na figura 26 vemos o rosto de John Hammond coberto pelas suas mãos, nesse caso o programa não foi capaz de identificar que se tratava de um rosto, nem ao menos o identificando, porém na mesma figura vemos o rosto de Ian Malcolm levemente distorcido por um efeito de câmara, porém neste caso o programa foi eficiente em identifica-lo.

Figura 27 - Rostos distorcidos ou encobertos por objetos Exemplo 01



Na figura 27 vemos novamente o programa não identificar o rosto de John Hammond, nessa imagem o mesmo tem um objeto na frente do rosto, na mesma imagem vemos novamente o rosto de Ian Malcolm ser corretamente reconhecido, mesmo sendo exibido de perfil.

Figura 28 - Rostos distorcidos ou encobertos por objetos Exemplo 02



Vemos na Figura 28 o rosto de John Hammond sendo corretamente reconhecido, porém neste exemplo o mesmo está em uma situação favorável para sua identificação.

Vemos também ao fundo o rosto de Ellie Battler não ser reconhecido pois as características básicas para que o programa reconheça que há um rosto estão encobertas.

Figura 29 - Rostos distorcidos ou encobertos por objetos Exemplo 03



Os testes acima mencionados foram realizados em uma máquina dotada de um processador i7 de 4ª Geração dual core com Clock médio de 2.4 GHz de processamento, 8 Gigabits de RAM e placa de vídeo dedicada da AMD Radeon M265. O Sistema Operacional foi o Ubuntu 18.04 LTS 64 bits. Foi utilizado o Python 3.6 nativo do Ubuntu antes referenciado.

O tempo de execução médio das dezesseis imagens usadas como exemplo foi de dez segundos, sendo que imagens acima de 500 Kbps não foram suportadas pelo computador. Segue abaixo gráfico com os tempos de execução de cada imagem.

Tabela 1 - Tempos de execução das imagens de teste

Figura	Tamanho da imagem em KB	Tempo de execução em segundos
14	333	9,323
15	420	11,759
16	324	9,071
17	432	12,095
18	321	8,987
19	345	9,659
20	365	10,219
21	339	9,491
22	398	11,143
23	470	13,159
24	342	9,575
25	364	10,191
26	376	10,527
27	355	9,939
28	341	9,547
29	441	12,347
Media	372,875	10,439

5 Considerações Finais

O reconhecimento de imagens está a cada dia mais presente em nossa rotina, e com essa avanço temos o uso constante da biometria para garantir a segurança de acesso e identificação, dessa forma tecnologias como o reconhecimento biométrico, de fala e o reconhecimento facial estão sendo amplamente utilizados, tornando assim a pesquisa e desenvolvimento do tema algo vital para o avanço da tecnologia.

No meio acadêmico o processamento de imagens através do estudo da inteligência artificial com ferramentas de deep learning estão abrindo portas para avanços incalculáveis. Socialmente já existem discussões morais e sociais a respeito do tema, tamanho seu impacto na tecnologia como a conhecemos, dessa forma abordar o tema e desenvolver novas tecnologia faz com que a evolução continue e passe a ser conhecida por todos.

Neste trabalho buscamos expor de forma sucinta o uso da inteligência artificial e deep learning para realizar o reconhecimento facial, explicando quais são as tratativas feitas nas imagens e suas bases. Apresentamos ferramentas de desenvolvimento gratuitas no mercado e através delas desenvolvemos um programa capaz de aprender a reconhecer faces e a diferencia-las umas das outras, com eficiência consideravelmente boa, tendo em vista a simplicidade do programa. Obtivemos algumas limitações referente ao tempo de execução do programa que teve um tempo médio de execução de 10 segundos, tempo que julgamos ser muito alto para uma aplicação comercialmente viável, porém tivemos uma boa surpresa em observar que mesmo em situação de imagens de baixa qualidade e posições das faces de forma a dificultar a identificação ainda assim o programa foi eficiente em identificar faces.

Consideramos que os objetivos estabelecidos no inicio desde trabalho foram atingidos, pois o programa foi eficiente para reconhecer faces em diversas imagens, com qualidade favorável e desfavoráveis e após o reconhecimentos foi possível diferencia-las umas das outras com uma boa precisão. Foram utilizadas as ferramentas propostas, com o uso do aprendizado de máquina e deep learning. A escolha por utilizar o Linux como plataforma e o Python como linguagem de desenvolvimento foi um grande acerto, visto que a linguagem possui uma grande quantidade de bibliotecas de código aberto para auxiliar o desenvolvimento, além de uma comunidade participativa e disposta a compartilhar conhecimento.

Uma sugestão para trabalhos futuros é tentar integrar um servidor de banco de dados nesta solução. Assim, não seria necessário buscar as imagens para o treinamento do programa, todo e qualquer ator e atriz, assim como outras pessoas cadastradas uma vez, todos os outros conectados ao mesmo banco e servidor poderiam treinar instantaneamente seus programas para a detecção das pessoas desejadas.

6 Referências Bibliográfica

- A. K. Jain, P. Flynn, and R. A. A., Handbook of Biometrics. <http://www.springer.com/us/book/9780387710402>: Springer US, 2008.
- RAVINDRA, S. What is image recognition and why is it used? 2017.
- BISHOP, C. M. Neural Networks for Pattern Recognition. Oxford: Clarendon Press, 1995.
- JEONG, H. (2012), "A comparison of the influence of electronic books and paper books on reading comprehension, eye fatigue, and perception", The Electronic Library, Vol. 30 No. 3, pp. 390-408.
- RUSLING, David. The linux kernel. Free digital book, 1996. Disponível em <http://rainbow.mimuw.edu.pl/SO/Linux-doc/LinuxKernel.ps.gz>.
- RICH, E., KNIGHT, K. Inteligência Artificial, Makron, 1993
- RABUSKE, R. Inteligência Artificial. Florianópolis: UFSC, 1995.
- LUGER, George F. Inteligência artificial: estruturas e estratégias para a resolução de problemas complexos. 4. ed. Tradução Paulo Martins Engel. Porto Alegre: Bookman, 2004
- RÄTSCHE, G. A brief introduction into machine learning. In: 21st Chaos Communication Congress. [S.l.: s.n.], 2004. Citado na página 16.
- DOMINGOS, P. A few useful things to know about machine learning. Communications of the ACM, ACM, v. 55, n. 10, p. 78–87, 2012. Citado na página 17.
- DUDA, R. O.; HART, P. E.; STORK, D. G. Unsupervised learning and clustering. Pattern classification, Wiley New York; Chichester, p. 519–598, 2001. Citado na página 17
- SATHYA, R.; ABRAHAM, A. Comparison of supervised and unsupervised learning algorithms for pattern classification. Int. J. Adv. Res. Artificial Intell, Citeseer, v. 2, n. 2, p. 34–38, 2013. Citado na página 17
- MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. Foundations of machine learning. [S.l.]: MIT press, 2012. Citado na página 1
- LI DENG, Dong Yu, et al. Deep learning: methods and applications. Foundations and Trends in Signal Processing, 7(3–4):4–6, 2014. Citado página 9
- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning. MIT Press, 2016. Citado na página 9

YOSHUA, Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016. <http://www.deeplearningbook.org>

YANG, Ming-Hsuan; KRIEGMAN, D.; AHUJA, N., 'Detecting faces in images: a survey', IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 1, pp. 34-58, 2002. Citado nas páginas 9 e 10

DEVIS S.; MANE, P. K.; AJAYKUMAR, “Face Detection System using OpenCV on Beagle Board”, 2015. Citado na página 10

VIOLA, P.; JONES M., 'Rapid object detection using a boosted cascade of simple features', Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001. Citado na página 10

ARAÚJO, G., 'Algoritmo para reconhecimento de características faciais baseado em filtros de correlação', Mestrado, UFRJ/COPPE, 2010. Citado na página 10

KING, D., “High Quality Face Recognition with Deep Metric Learning”, 2017. Citado na página 12

HUANG, G. B.; RAMESH, M.; BERG, T.; LEARNED-MILLER, E., (2007) “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments”, University of Massachusetts, Amherst, Technical Report 07-49 Citado na página 12.

HAUGELAND, John. Artificial Intelligence: The Very Idea. Massachusetts: The MIT Press, 1985.