

Universidade Paulista - UNIP

Roberto De Michielli Gonçalves de Araújo

RECONHECIMENTO FACIAL NA SEGURANÇA PÚBLICA.

**Limeira
2021
Universidade Paulista - UNIP**

Roberto De Michielli Gonçalves de Araújo

RECONHECIMENTO FACIAL NA SEGURANÇA PÚBLICA.

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciências da computação sob a orientação do professor Me. Danilo Rodrigues Pereira.

Limeira
2021

Roberto De Michielli Gonçalves de Araújo

RECONHECIMENTO FACIAL NA SEGURANÇA PÚBLICA.

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciências da Computação sob a orientação do professor Me. Danilo Rodrigues Pereira.

Aprovada em XX de XXXXX de 201X.

BANCA EXAMINADORA

Prof. Dr. Nome completo

Prof. Me. Nome completo

Prof. Esp. Nome completo

DEDICATÓRIA

Dedico este trabalho a minha mãe
Luciane, minha namorada Giovanna, minha
tia Eliane por me apoiarem.

*“O sucesso nada mais é que ir de fracasso
em fracasso sem que se perca o entusiasmo”.*

(Winston Churchill)

RESUMO

Texto em parágrafo único, no máximo 500 palavras...

Palavra-Chave: Reconhecimento facial, Haarcascade, OpenCV, Python, Segurança Pública.

ABSTRACT

Text...

Keywords: Facial recognition, Haarcascade, OpenCV, Python, Public security.

LISTA DE FIGURAS

Figura 1 – Tipos de features.....	17
Figura 2 – Descriptor LBPH.....	19

Figura 3 – Matriz.....	20
Figura 4 – Reconhecimento Facial (Fonte Google Imagens).....	21

LISTA DE QUADROS

Nenhuma entrada de sumário foi encontrada.

LISTA DE ABREVIATURAS

OpenCV

Open Source Computer Vision - (Visão computacional de código aberto)

LGPD	Lei Geral de Proteção de Dados Pessoais
LBPH	<i>Local Binary Patterns Histograms</i> - (Histogramas de Padrões Binários Locais)
VPN	<i>Virtual Private Network</i> – (Rede Virtual Privada)

SUMÁRIO

1. INTRODUÇÃO.....	12
--------------------	----

1.1	Objetivo.....	13
1.2	Justificativa.....	13
1.3	Metodologia.....	13
2.	RECONHECIMENTO FACIAL.....	14
2.1	Tecnologia.....	14
2.2	Bibliotecas.....	15
2.2.1	OpenCV.....	16
2.2.2	Haarcascade.....	16
2.2.3	Local Binary Patterns Histograms (LBPH).....	18
2.2.4	NumPy.....	19
2.3	Segurança Pública.....	20
2.4	Aplicação e Regulamentação.....	21
3.	AMBIENTE DE PROGRAMAÇÃO.....	22
3.1	Preparando o Interpretador.....	22
3.2	PIP.....	22
3.3	Python.....	22
3.4	PyCharm.....	23
4.	DESENVOLVIMENTO.....	23
4.1	Algoritmo.....	24
4.2	Códigos.....	24
	CONCLUSÃO.....	25
	REFERÊNCIAS BIBLIOGRÁFICAS.....	26

1. INTRODUÇÃO

De fato, a tecnologia é impressionante, a alguns anos atrás tínhamos que nos virar com carros, celulares, equipamentos eletrônicos de baixa qualidade comparado aos de hoje em dia, para nós isso pode até ter mudado muito, porém aos policiais/bombeiros/militares que estão em combate e que precisam de um suporte maior não mudou quase nada.

Com uma pesquisa rápida no google sobre a criminalidade no Brasil, é possível ver o quanto nosso país é violento. Em 17/06/2020, a g1 publicou em seu site durante a pandemia o aumento na taxa do número de assassinatos no Brasil, em 29/11/2019, cinco meses antes da publicação do g1, o site “Agencia Brasil”, publicou em seu site em que um homem fez seis reféns em um bar no Rio de Janeiro. O homem estava armado com uma faca, o que minimiza o risco de mais mortes no local, levando em consideração o armamento da força operacional que estava no local, porém o avanço do BOPE ainda estava complicado, a vida de reféns ainda estava em grande risco, qualquer “gatilho” poderia fazer com que o criminoso pudesse executar um dos reféns.

Agora imagina poder estar no local, dentro do bar, analisar o ambiente, e conseguir descobrir quem realmente estava lá dentro, levando em consideração que poderia ter mais de um criminoso, qual armamento ele possui, conseguir encontrar um momento perfeito, em que o criminoso esteve longe dos reféns e conseguir prosseguir em segurança com a invasão ao local.

utilizando reconhecimento facial poderia ser uma ótima opção para momentos, saber a posição e quantas pessoas tem no local, analisar o ambiente, e descobrir os exatos possíveis riscos do momento, e ainda sem precisar colocar a vida de ninguém em risco.

Levando em consideração a variedade de operações que este dispositivo pode oferecer,

O equipamento visa auxiliar-

1.1 Objetivo

O objetivo desse trabalho é realizar pesquisas direcionada ao reconhecimento facial com a biblioteca OpenCV (Python) demonstrando sua eficiência e limitações na utilização na segurança pública, no final do trabalho será analisado a possibilidade do uso de reconhecimento facial na identificação de criminosos foragidos auxiliando no combate de crimes.

Será desenvolvido um protótipo de um software de reconhecimento biométrico facial e realizar testes em diferentes cenários para mostrar a efetividade da biblioteca OpenCV, onde serão analisados os testes comparativos com base nos diferentes cenários.

Este trabalho visa criar um algoritmo para a programação do reconhecimento biométrico facial, onde será possível entender e compreender como o software realiza a identificação de faces, onde e por quem pode ser utilizado esse recurso com base na lei LGPD.

1.2 Justificativa

No Brasil existem mais de 20,4 mil foragidos e 353,2 mil mandados de prisão pendentes de cumprimento (Conselho Nacional de Justiça, 2019), são vários criminosos que estão soltos nas ruas, cometendo crimes e prejudicando a população, o sistema de reconhecimento facial na segurança pública, permite que seja realizado a identificação dos rostos das pessoas, auxiliando no combate ao crime e na redução de foragidos.

Este trabalho visa preencher essa falha que temos na segurança pública, existem muitos criminosos foragidos nas ruas, e continuaram até que sejam reconhecidos e presos. O sistema de reconhecimento facial irá auxiliar e melhorar a identificação de criminosos foragidos para que sociedade se torne mais segura.

1.3 Metodologia

A primeira fase do trabalho consiste em realizar pesquisas sobre a tecnologia do reconhecimento facial na linguagem Python utilizando a biblioteca OpenCV e sobre as demais bibliotecas, com o objetivo de compreender suas tecnologias e aplicações. Utilizando essa base, é plausível perceber como é realizado a captura dos pontos nodais da face de uma pessoa, e onde e por quem pode ser implantado respeitando a lei LGPD.

Após isso será realizado testes de eficiência em diferentes cenários para compreender como ter um excelente diretório de fotos no banco de dados com o intuito de melhorar a porcentagem de acerto visando a melhoria da segurança pública e otimizar as operações de forças especiais em aeroportos ou grande eventos como as olimpíadas, com isso realizar o auxílio no reconhecimento de criminosos que estão foragidos nas “ruas”, cujo objetivo é diminuir a impunidade e os criminosos soltos em nossa sociedade.

2. RECONHECIMENTO FACIAL

O reconhecimento facial é um tipo de biometria, porém ao invés da digital, é no rosto onde é realizado a detecção dos pontos de referência, através de um processamento computacional usando a imagem capturada pela câmera que é comparada com a foto no banco de dados, mas para realizar essa comparação, deve-se definir os traços únicos da face da pessoa e codificar em binário.

Os pontos nodais são pontos de referência no rosto de uma pessoa, temos cerca de 80 pontos nodais. Os pontos de referência podem ser, por exemplo a distância entre os olhos, o tamanho do queixo e nariz, o formato do lábio e mandíbula (NSTC Subcommittee, 2006, p. 1). A extração da assinatura facial só estará completa após a captura de cada ponto nodal armazenado no banco de dados.

2.1 Tecnologia

O software antes de realizar a extração de cada ponto nodal, é realizado a busca da imagem do rosto da pessoa, onde será localizado a exata posição dos pontos nodais. Após esse procedimento, será analisado os pontos nodais, que precisam estar na mesma posição das características do rosto, olhos, mandíbula da imagem capturada. A análise da imagem é feita com base em uma imagem que pode ser tanto uma foto ou um vídeo, capturada por qualquer tipo de câmera, seja ela do celular, de segurança ou webcam ou já armazenada. Após essa etapa, o software aplica uma sequência de filtros para rastrear as características humanas e diferenciar o ambiente da pessoa, na qual se assemelham os olhos, nariz, mandíbula e boca.

Existem uma grande diferença entre detectar um rosto e a extração dos pontos nodais, a detecção do rosto o software apenas reconhece que existe um rosto na imagem, já a extração dos dados do rosto, é o software realizando o reconhecimento facial, calculando os diferentes detalhes e analisando os pontos únicos que cada pessoa possui.

A técnica de biometria facial trás diversos benefícios para quem adquirir essa tecnologia tão eficiente, como por exemplo, melhorias na segurança de uma empresa, facilidade no processo de integração do sistema, porcentagem de precisão alta, mais automação e combate a fraudes de falsificação de documentos. Mas para extrair o máximo desse recurso é necessária uma câmera de alta resolução e imagens nítidas, caso o contrário pode haver falha no reconhecimento, informando outra pessoa da imagem analisada.

2.2 Bibliotecas

Em desenvolvimento de softwares, alguns recursos com API (biblioteca de funções) são bastante utilizados para aumentar a produtividade e qualidade dos softwares criados, são uma forma eficiente de diminuir o tempo no desenvolvimento e aumentar a produtividade na construção do código.

As APIs, possuem diversos conjuntos de funções e módulos prontos para serem usados pelo desenvolvedor. Em Python, existem mais de 137.000 (cento e trinta e sete mil) bibliotecas a sua disposição.

A grande vantagem de se utilizar as bibliotecas é que se pode usa-las em diversos projetos diferentes, no caso desse projeto, foram utilizados quatro projetos, “Captura.py, Treinamento.py, AjusteTreinamento.py e ReconhecimentoProfundo.py”. As bibliotecas utilizadas nesse trabalho foram: OpenCV, haarcascade, Ibph, numpy, shutil, paths. Para realizar a instalação dessas bibliotecas, é necessário possuir o pip instalado em sua máquina, pois é ele quem irá instalar em seu ambiente de programação as bibliotecas de programação, caso já o tenha instalado, é necessário atualiza-lo para a última versão lançada, evitando assim erros e conflitos de versões entre o pip e as bibliotecas.

2.2.1 OpenCV

Open Source Computer Vision (OpenCV), é uma biblioteca para o processamento de imagens. A biblioteca tem como foco o monitoramento de funções sobre a visão do computador. OpenCV é uma das bibliotecas mais fáceis de

se aprender, por conta de sua documentação ser “forte”, contudo, a biblioteca possui grande diversidade de funções nas quais com elas se aprende de forma eficiente sobre a visão computacional.

OpenCV é capaz de gravar e ler as imagens (fotos ou vídeos) ao mesmo tempo, este pacote permite a compreensão de um ambiente de 3 dimensões pela perspectiva de um ambiente de 2 dimensões, além de ser possível capturar em qualquer momento um quadro do vídeo (~~da imagem~~), analisar suas propriedades como o plano de fundo, movimentos ou qualquer objeto que esteja nas imagens, como por exemplo carros, placas, rostos, olhos, pontos nodais, copos, etc.

No reconhecimento facial, o OpenCV será utilizado para fazer a captura, análise e o reconhecimento de objetos nas imagens, sejam fotos ou vídeos, no caso, o rosto.

2.2.2 Haarcascade

Haarcascade é um algoritmo executado em tempo real que pode realizar a detecção de objetos em imagens, não importando posição ou tamanho. O código é o mais popular entre os algoritmos de reconhecimento de objetos no OpenCV, é também, um importante parte da visão computacional e processamento de imagens, que possibilita o treinamento de seus detectores, para realizar o reconhecimento de faces, carros, placas e até de animais. Um dos seus principais benefícios é sua eficiência e velocidade na detecção de objetos em tempo real, porém uma desvantagem são seus possíveis reconhecimentos de falsos positivos, que necessitam ajuste nos parâmetros de seus detectores.

O algoritmo do haarcascade funciona com base em cálculos em regiões retangulares ao lado de locais específicos. O cálculo é realizado na diferença da soma e da soma das intensidades dos pixels em cada local e região da imagem. São 4 regiões retangulares no total em que o código realiza seus cálculos, são eles, edge, line, center-surround e four-retangle.

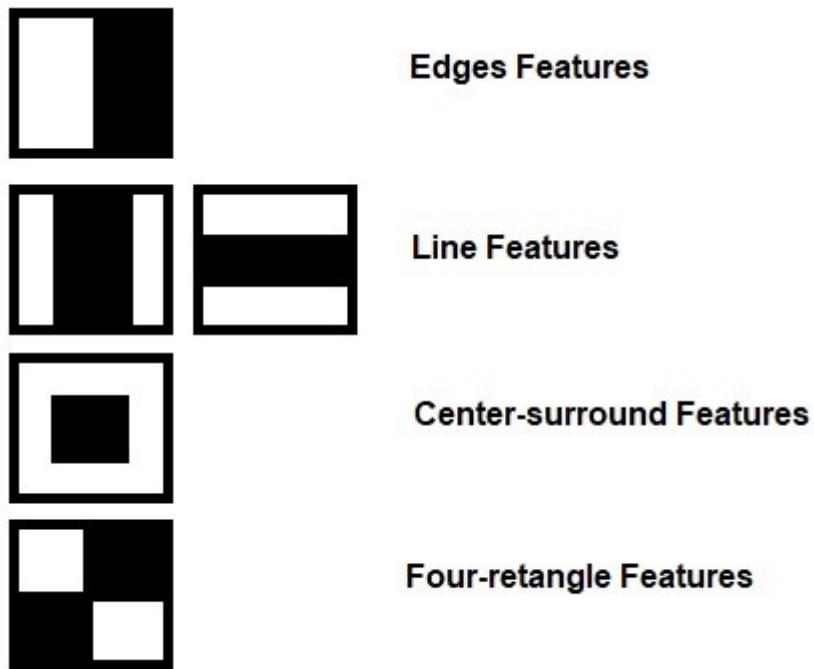


Figura 1 – Tipos de features.

O código os executa para reconhecer o objeto a partir das variações da intensidade de luz nos pixels, na qual, é gerado vários classificadores, um para cada feature, é seguido um passo a passo para o algoritmo conseguir encontrar o objeto, percorrendo a imagem toda, executando pequenos classificadores em “cascatas”. Primeiramente é definido um tamanho de janela e escala os features, varrendo a imagem, em cada uma dessas janelas, onde são executados os pequenos classificadores com base nos valores dos pixels daquela janela, caso o código retorne positivo, volta a percorrer a imagem com os pequenos classificadores e se não houver mais classificadores para serem executados, é concluído que existe um objeto na imagem, mas, caso retorne falso, o código pula para a próxima janela, e se não houver mais janelas, o resultado é que não há objetos na imagem.

2.2.3 Local Binary Patterns Histograms (LBPH)

Local Binary Patterns Histograms ou LBPH se baseia em retirar informações locais dos pixels da imagem. O algoritmo não analisa a imagem inteira de uma vez só, mas apenas aponta as informações de locais de um objeto. As informações

extraídas possuem dimensões pequenas, o que se torna o processo mais poderoso para a análise e reconhecimento de objetos.

LBPH realiza o cálculo com base na representação local da textura da imagem, esta representação é feita com comparações por cada pixel ao seu redor, do qual está sendo analisado, e esses cálculos geram os padrões binários, onde são utilizados para dar características a textura da imagem e suas principais características com base em seus padrões, processando os pixels locais, gerando um descriptor de textura da imagem mais potente.

Os padrões binários são para definir a textura e os padrões dos objetos de uma imagem, tornando a imagem mais robusta, com uma textura mais poderosa. Para gerar um descriptor, é necessário converter a imagem capturada em escalas de cinza, selecionamos um pixel central em um local na imagem, e um valor LBPH é calculado e armazenado nesse pixel central em uma matriz de duas dimensões (2D) de saída com as mesmas dimensões da imagem de entrada, na figura abaixo, é ilustrado como é feito esse cálculo em um local 3x3 da imagem.

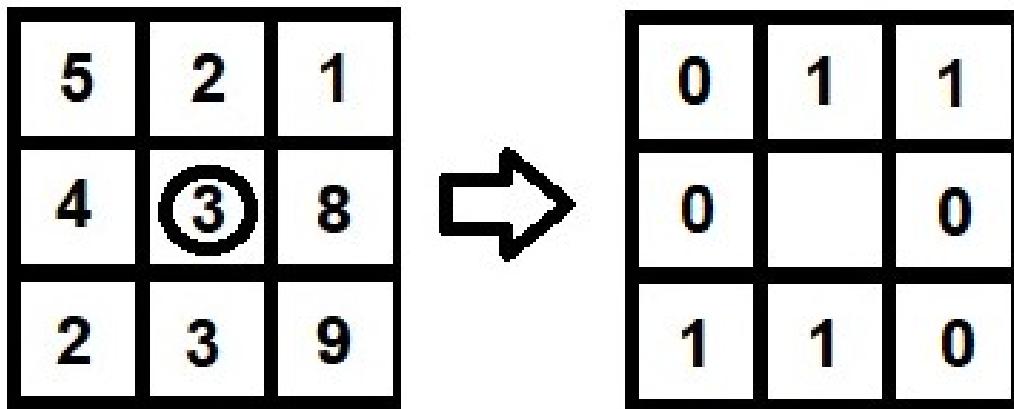


Figura 2 – Descriptor LBPH.

Caso o valor, no caso da imagem, a intensidade do pixel central for maior ou igual a intensidade de seu vizinho, o valor definido é 1, se não, é definido para 0. Na

figura acima, possui um total de 8 pixels ao redor do pixel central, $2^8 = 256$ possíveis de código LBPH.

2.2.4 NumPy

A biblioteca numpy é muito usada por desenvolvedores que precisam realizar grandes cálculos numéricos, tornando-se uma forte biblioteca para cálculos em matrizes e matrizes multidimensionais. O numpy é constantemente usado em *machine learning* e computação gráfica.

As imagens em um computador são construídas com bases em matrizes multidimensionais, a própria biblioteca do numpy fornece diversas funções para tratamento e manipulação de imagens.

Os algoritmos de machine learning, precisam de muitos cálculos de numéricos em matrizes, adição, multiplicações de matrizes, transposições. Para armazenar os dados de parâmetros, treinamentos de algoritmos de machine learning, é utilizado os arrays numpy.

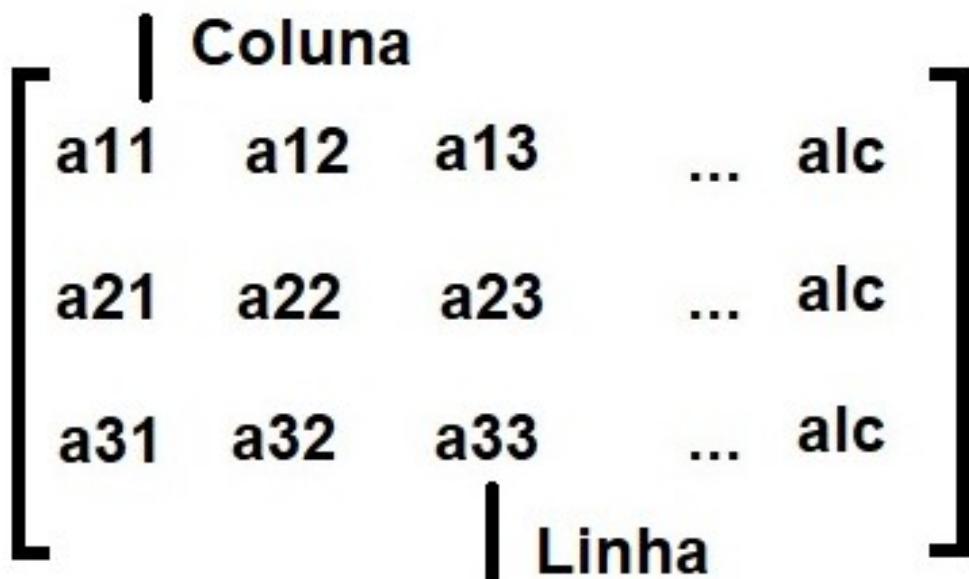


Figura 3 – Matriz.

2.3 Segurança Pública

Uma pesquisa realizada pela *Surfshark* (empresa dedicada a serviços de VPN) revelou que existem mais de 97 países que utilizam os recursos de reconhecimento biométrico facial em diversos tipos de monitoramento, como escolas, bancos, hospitais e aeroportos e com uma estimativa do departamento de segurança interna dos estados unidos, que até 2023 mais de 96% de companhias aéreas examine seus passageiros, o que para um humano seria impossível de realizar, identificar criminosos e foragidos no meio de centenas de pessoas usando serviços como aeroportos e metros, seria como procurar uma agulha num palheiro.

A tecnologia de reconhecimento facial veio para simplificar e otimizar os serviços de segurança e monitoramento em diversos tipos de locais onde terroristas, fugitivos e criminosos podem estar, com apenas um simples “gatilho”, como uma imagem do rosto do suspeito, o sistema de reconhecimento biométrico facial detecta quem é a pessoa em questão na imagem, auxiliando assim, a segurança de uma cidade inteira, em pontos públicos, priorizando a privacidade dos cidadãos.

Infelizmente pode acontecer do sistema errar no reconhecimento do indivíduo, isso pode acontecer devido ao treinamento incorreto e registros no banco de dados serem de baixa qualidade. Por isso é essencial que o treinamento seja feito de maneira correta, e as imagens serem de alta resolução.

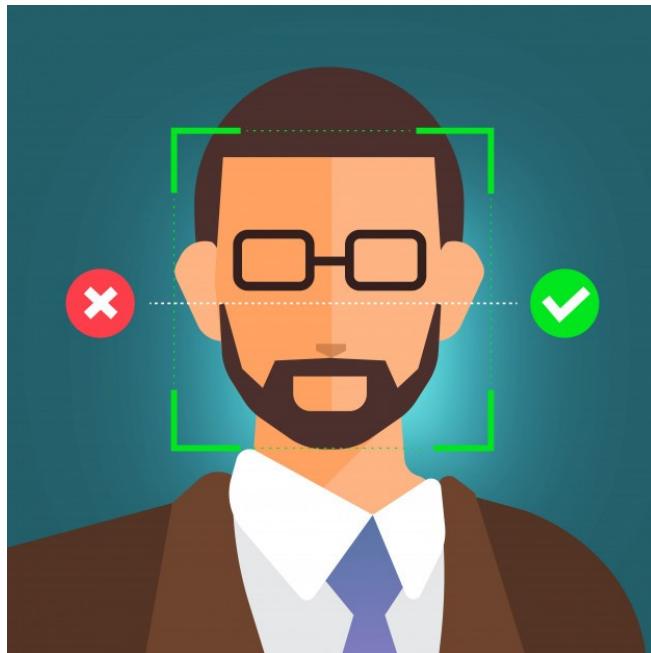


Figura 4 – Reconhecimento Facial (Fonte Google Imagens)

2.4 Aplicação e Regulamentação

O sistema de reconhecimento biométrico facial pode ser aplicado em muitos lugares pela cidade, em locais onde existe muita aglomeração, como centro da cidade, mercados, academia, shoppings, entre muitos outros. Esse sistema pode ser usado por qualquer empresa, e todas elas com o mesmo objetivo, melhorar a segurança, identificar pessoas no ambiente, barrar a entrada de pessoas não autorizadas ou permitir as que estão.

O sistema de reconhecimento facial pode ser usado para melhorar nossa segurança, a identificação de criminosos se torna muito mais eficiente, pois os criminosos são reconhecidos se estiverem no alcance das lentes de segurança do sistema de reconhecimento biométrico facial, utilizando os registros em seu banco de dados contendo os dados sobre o criminoso, sua captura se torna menos complicada, com esse sistema é possível prevenir que mais crimes aconteçam.

Segundo a Lei Geral de Proteção de Dados Pessoais (LGPD), **Lei nº 13.709, de 14 de agosto de 2018**, define que:

“Esta Lei dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o

objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural. (Brasil, 2018).“

Existe a proteção do tratamento de dados pessoais, e como o reconhecimento facial trata o rosto da pessoa como um dado, a lei entra em vigor, entretanto, no artigo 11, II, alínea "e", da LGPD: "*e) proteção da vida ou da incolumidade física do titular ou de terceiro;* (Brasil, 2018, p. 1).", caso o estabelecimento solicite ao poder público uma autorização, especificando o objetivo e a finalidade do uso desses dados, é possível utilizar o banco de dados de foragidos e bandidos em seu sistema de reconhecimento biométrico facial.

3. AMBIENTE DE PROGRAMAÇÃO

Texto

3.1 Preparando o Interpretador

Texto

3.2 PIP

Texto

3.3 Python

A melhor linguagem de programação para criar um software de reconhecimento facial, é Python, cujo o mesmo é nomeado por quatro anos seguidos como melhor linguagem de programação pelo TIOBE (*lista ordenada e classificada pela frequência e popularidade de pesquisa na internet*).

Python possui código aberto, além de gratuito, tanto os desenvolvedores quanto a comunidade trabalham juntos para melhorar a linguagem, trazendo consigo diversos recursos e benefícios. Python é relativamente a linguagem mais fácil e

acessível para aprender além de oferecer oportunidades profissionalizantes com alto rendimento, a linguagem também oferece uma vasta e ampla biblioteca para as demais utilidades, sejam elas reconhecimento facial (OpenCV), inteligência artificial (TensorFlow).

Nos dias de hoje, reconhecimento facial e Python são muito utilizados juntos, um código de nível alto consegue suportar diversos estilos, processuais, funcionais e orientados a objeto, apresenta suporte para rede neural reconhecimento de movimento e imagens. Python é a melhor escolha de linguagem de programação, passando Java e c++, pois possui mais recursos, é mais rápido e fácil que ambos.

3.4 PyCharm

O IDE PyCharm é um ambiente de desenvolvimento para a programação e a linguagem Python, onde é disponibilizado vários e diversos recursos que facilitam o desenvolvimento de software, o mesmo pode ser comparado com a IDE padrão do Python.

Este trabalho foi utilizado o PyCharm, para facilitar o desenvolvimento do software de reconhecimento facial e os testes de desempenho da biblioteca OpenCV, a versão da IDE utilizado foi a 2021.1.1, para o sistema operacional Windows 10, disponível em no site oficial do JetBrains.

4. DESENVOLVIMENTO

Durante o desenvolvimento desse trabalho e do software, foi possível compreender e entender como é feito e realizado o reconhecimento facial de uma pessoa, o que o software busca na imagem e como consegue interpretar e realizar os cálculos necessário para identificar a face e reconhece-la no banco de dados.

Foi necessário compreender e realizar pesquisas para entender quais seriam as melhores e necessárias bibliotecas para poder desenvolver o código e realizar a detecção e reconhecimento facial para que seja possível a utilização desse sistema na segurança publica respeitando a lei LGPD.

4.1 Algoritmo

Texto...

Basta tratarmos a imagem e extrair esses pontos nodais e busca-los no banco de dados. O software de reconhecimento facial não é diferente, busca na imagem em escala de cinza, em locais específicos na imagem, os pixels centrais e os que estão ao redor, sempre identificando os pontos nodais através das bibliotecas responsáveis por esse tratamento na imagem, como explicado nos tópicos de reconhecimento facial.

4.2 Códigos

O código foi baseado nos códigos do github do ..., como base, foi possível desenvolver o código a seguir:

CONCLUSÃO

Finalmente....

REFERÊNCIAS BIBLIOGRÁFICAS

- HEINEN, M. R. **Autenticação de Assinaturas Online utilizando Redes Neurais Artificiais.** São Leopoldo: Universidade do Vale do Rio dos Sinos, 2002.
- KSHIRSAGAR, V.P., BAVISKAR, M. E. **Face Recognition using Eigenfaces.** 3rd International Conference on Computer Research and Development (ICCRD). 2011.
- LI, S.; JAIN, A. **Handbook of face recognition.** Springer, 2005. (Springer eBooks collection: Computer science).
- OPENCV. 2016. **Face Recognition with OpenCV.** Disponível em http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#face-recognition-with-opencv. Acesso em: 01 mar. 2021.
- YANG, M. H.; AHUJA, N. & KRIEGMAN, D. **Detecting Faces in Images: A Survey.** IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 24, n. 1, pp. 34-58, 2002.
- AGARWAL, M.; AGRAWAL, H.; JAIN, N.; KUMAR, M. **Face Recognition Using Principle Component Analysis, Eigenface and Neural Network.** Signal Acquisition and Processing, 2010.
- BOECHAT, G. C. **Investigação de um Modelo de Arquitetura Biometrica Multimodal para Identificação Pessoal.** Universidade Federal do Pernambuco, 2008.
- BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV.** O'Reilly Media, 2008
- CARNEIRO, Larissa Natália das Virgens. **Reconhecimento de Face Invariante a Iluminação baseado em uma Abordagem Supervisionada.** Universidade Federal de Ouro Preto, 2012.
- LIU, S.; Silverman, M. **A Practical Guide to Biometric Security Technology.** IEEE Computer Society, 2001.
- MARENCONI, Maurício; STRINGHINI, Denise. **Tutorial: Introdução à Visão Computacional usando OpenCV.** RITA, 2009.

NARANG, Sudha; JAIN, Kriti; SAXENA, Megha; ARORA, Aashna. **Comparison of Face Recognition Algorithms Using Opencv for Attendance System.** International Journal of Scientific and Research Publications, 2018.

OPENCV. **Opencv.** Disponível em <<https://opencv.org>>. Acesso em: 06 abr. 2021.

ZHAO, W.; CHELLAPPA, R.; PHILLIPS, J.; ROSENFELD, A. **Face recognition: a literature survey.** ACM Computing Surveys, vol. 35, no. 4, 2003.

VIOLA, Paul.; JONES, Michael. **Rapid Object Detection using a Boosted Cascade of Simple Features.** Kauai, OI, EUA, IEEE Computer Society, 2001.

Brasil. LEI Nº 13.709, 14 ago. 2018. **Lei Geral de Proteção de Dados Pessoais (LGPD).** Disponível em:

http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/Lei/L13709.htm. Acesso em: 11 maio. 2021.

Conselho Nacional de Justiça. **Ministério da Justiça e Segurança Pública e Conselho Nacional de Justiça assinam acordos de cooperação técnica.** Disponível em Ministério da Justiça: <https://www.justica.gov.br/news/collective-nitf-content-1567522061.62>. Acesso em: 19 maio. 2020.

NSTC Subcommittee. **Face Recognition.** (N. S. (U.S.), Editor). disponível em Homeland Security Digital Library: <https://www.hSDL.org/?abstract&did=463904> Acesso em: 19 maio. 2021.

Portanto, o reconhecimento facial pode ser algo simples em sua ideia, pois nada mais é que um equipamento com capacidade de reconhecer e identificar rostos, mas sua funcionalidade, versatilidade e utilidade é de grande ajuda, portanto, é importante mostrar de forma eficiente o quanto útil um sistema com reconhecimento facial pode ser em casos onde existem procurados em nossa sociedade ou até mesmo identificar inocentes que foram sequestrados, poder analisar o ambiente, entender o risco, saber com quem está lidando, sem precisar estar no local e sem precisar colocar a vida de ninguém em risco, é incrível.