**CENTRO PAULA SOUZA**

**ETEC VASCO ANTONIO VENCHIARUTTI – JUNDIAÍ - SP**

**TÉCNICO EM INFORMÁTICA PARA INTERNET**

Artigo desenvolvido na disciplina de Planejamento de Trabalho de Conclusão de Curso sob orientação dos professores Renan Aleixo Paganatto e Robson Alberghini.

**RAF (Reconhecimento Automático de Faces)**

**Reconhecimento facial por meio de processamento e comparação de imagens em tempo real.**

Higor Bernardes da Silva

Isabela Leme Sena

João Pedro Barsotti

João Victor Reis de Paula

Whillian Gabriel Faustino Silva

# RESUMO

O objetivo do trabalho aqui apresentado foi desenvolver uma ferramenta que faça a captura de faces com o uso de uma webcam e assim analisá-las e conferir se estão cadastradas ou não em nosso banco de dados. Foram utilizados diversos conceitos como linguagem de programação Python, bibliotecas dessa linguagem e outras ferramentas analisadas para fazer com que nosso projeto fosse executado da maneira mais eficiente possível. Uma coleta de informações foi necessária, pois assim entenderíamos as origens desse tema e como são utilizadas outras ferramentas com funcionalidades de reconhecimento facial. Toda essa pesquisa tornou possível a criação e o funcionamento do projeto que tem como foco a segurança.

**Palavras-chave:** Reconhecimento, Faces, Python, Bibliotecas, Webcam.

# INTRODUÇÃO

O uso da visão computacional abrange uma variedade de campos da tecnologia, como reconhecimento de faces, caracteres e objetos sendo eles animados ou inanimados, na maioria dos casos aplicados em imagens das mais diversas. Neste projeto o uso da tecnologia visa identificar faces e retornar em frame os nomes dos usuários cadastrados, caso contrário retornará o valor desconhecido.

A visão computacional, em conjunto com a programação em linguagem *Python*, está sendo usada em diversos campos atualmente, como reconhecimento de pessoas, de padrões e no reconhecimento de caracteres [...]. (LEITE; ANTONELLO, 2017, p.1).

Segundo Antonello (2017), dos cinco sentidos dos seres humanos a visão é enfatizada como sendo uma das mais importantes, senão a mais. É por isso que "dar" esse significado às máquinas produz resultados impressionantes, já que as imagens estão em todos os lugares e a capacidade de reconhecê-las torna as máquinas mais eficientes.

O principal objetivo do projeto é identificar usuários através do seu rosto e retornar se o indivíduo está armazenado em nossa *engine*.*py*, que é um arquivo em *python* que funciona como um banco de dados controlando os dados do usuário que estiverem armazenados. O atual projeto pode ser utilizado em diversos ramos, principalmente na segurança de apartamentos ou condomínios.

O projeto utiliza de uma *webcam* com resolução 1080p rodando a 30 fps e um sistema operacional *Windows* 10 padrão. Os códigos são desenvolvidos utilizando a linguagem de programação *Python* e o editor de códigos fontes *VSCode* com as bibliotecas ***OpenCV*, *Face\_Recognition*, *PIL*, *Cmaker*, *CVzone*, *Numpy*, *Dlib*** entre outras, sendo elas as mais importantes.

Buscando melhorar a eficiência da visão computacional em diferentes ambientes acabamos percebendo que apenas uma biblioteca por mais bem desenvolvida que seja não consegue grandes resultados individualmente, porém quando aplicada a um conjunto de bibliotecas corretamente utilizadas pode alcançar até 99% de precisão no quesito reconhecimento de faces, transformando o projeto em uma ferramenta bastante promissora.

# DESENVOLVIMENTO

Com o objetivo de uma boa organização de tarefas, dividimos o projeto em etapas. Foi feito um estudo das bibliotecas para nos introduzirmos nesse tema e compreendermos sobre como é feito e como podemos utilizar das suas ferramentas, junto da pesquisa nos deparamos com a linguagem de programação Python.

A primeira etapa foi encontramos a linguagem de programação que iríamos utilizar, após muita pesquisa foi definido que a linguagem Python era a mais eficiente muito por causa das suas bibliotecas, possibilidades e desenvolvimento, mas antes de tudo vamos definir o que é Python e quem o criou, segundo o site (Wikipédia):

* ***Python***

É uma [linguagem de programação de alto nível](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_de_alto_n%C3%ADvel), [interpretada](https://pt.wikipedia.org/wiki/Linguagem_interpretada) de [script](https://pt.wikipedia.org/wiki/Linguagem_de_script), [imperativa](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_imperativa), [orientada a objetos](https://pt.wikipedia.org/wiki/Orienta%C3%A7%C3%A3o_a_objetos), [funcional](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_funcional), de [tipagem](https://pt.wikipedia.org/wiki/Tipo_de_dado) dinâmica e forte. Foi lançada por [Guido van Rossum](https://pt.wikipedia.org/wiki/Guido_van_Rossum) em [1991](https://pt.wikipedia.org/wiki/1991). Atualmente, possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela [organização sem fins lucrativos](https://pt.wikipedia.org/wiki/Organiza%C3%A7%C3%A3o_sem_fins_lucrativos) [Python Software Foundation](https://pt.wikipedia.org/wiki/Python_Software_Foundation). Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem, como um todo, não é formalmente especificada. O padrão [de fato](https://pt.wikipedia.org/wiki/De_facto) é a implementação [Python.](https://pt.wikipedia.org/wiki/CPython)

A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma [sintaxe](https://pt.wikipedia.org/wiki/Sintaxe) concisa e clara com os recursos poderosos de sua [biblioteca](https://pt.wikipedia.org/wiki/Biblioteca_de_rotinas) padrão e por [módulos](https://pt.wikipedia.org/wiki/M%C3%B3dulo_de_um_programa) e [frameworks](https://pt.wikipedia.org/wiki/Framework) desenvolvidos por terceiros.

Python é uma linguagem de propósito geral de [alto nível](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_de_alto_n%C3%ADvel), [multiparadigma](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_multiparadigma), suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens. Devido às suas características, ela é utilizada, principalmente, para processamento de textos, dados científicos e criação de [CGIs](https://pt.wikipedia.org/wiki/CGI) para páginas dinâmicas para a web. Foi considerada pelo público a 3ª linguagem "mais amada", de acordo com uma pesquisa conduzida pelo site [Stack Overflow](https://pt.wikipedia.org/wiki/Stack_Overflow) em 2018[[7]](https://pt.wikipedia.org/wiki/Python#cite_note-7) e está entre as 5 linguagens mais populares, de acordo com uma pesquisa conduzida pela [RedMonk](https://pt.wikipedia.org/wiki/RedMonk).

A segunda etapa foi conhecermos mais sobre as bibliotecas que utilizamos e aprendermos o que elas são, como elas funcionam, esta foi a pesquisa que demandou mais tempo e esforço, mas também foi aqui que entendemos a grandiosidade e as possibilidades que estas bibliotecas podem entregar. As mais importantes são:

* **OpenCV**

Começando pela *OpenCV* para a detecção de rostos, escolhemos utilizar essas ferramentas pois é biblioteca de código aberto para visão computacional, processamento de imagem e aprendizagem de máquina, sendo criada no final da década de 90 pela Intel, sua licença é do tipo BSD que é livre para uso acadêmico e comercial, possuindo interfaces C++, C, *Python* e *Java*, suportando sistemas operacionais como *Windows, Linux, Mac OS, IOS e Android*. Tem como eficiência computacional o foco em aplicações em tempo real (Leandro Passarelli, 2017).

O *OpenCV* trabalha atualmente com três algoritmos de reconhecimento facial, sendo eles *Eigenfaces, Fisherfaces e LBPH,* que estão presente no desenvolvimento desta aplicação.

# Face\_Recognition

O reconhecimento facial se tornou uma área popular de pesquisa em visão computacional é uma das aplicações mais bem-sucedidas de análise e compressão de imagens.  A face de uma pessoa é o essencial fator utilizado para reconhecê-la no dia-a-dia, ou seja, reconhecer um rosto é uma tarefa simples, possuímos habilidades capazes de guardar uma quantidade imensa de rostos em nossa memória e ainda assim reconhecer a maioria delas em uma pessoa em questão de décimos de segundos (Tsutsumi Kuroiwa & Carro, 2015).  Segundo Szeliski (2010), citado por Prado (2018, p. 41) “Os métodos de reconhecimento facial atuais trabalham melhor utilizando imagens faciais frontais e em condições de iluminação uniformes. Entretanto, bases de imagens reais coletadas podem incluir grandes variações de pose e de iluminação, as primeiras abordagens de reconhecimento facial conhecidas consistiam em localizar características distintas na imagem, como por exemplo os olhos, nariz e boca, e então medir as distâncias entre estes pontos de características. Algumas abordagens de reconhecimento facial mais recentes trabalham na comparação de imagens em escala de cinza, projetadas em subespaços de baixa dimensão”.

A tarefa de reconhecimento facial é composta por três processos distintos: Registro, verificação e identificação. Os processos se diferenciam pela forma de determinar a identidade de um indivíduo.

* **PIL**

Segundo o *Wikipédia* o *Python Imaging Librarym* (PIL) é uma biblioteca da linguagem de programação *Python* que adiciona suporte à abertura e gravação de muitos formatos de imagem diferentes.

# Cmake

De acordo com o *Wikipédia* o *Cross Plataform Make* (CMake) pode compilar código fonte, criar bibliotecas, gerar empacotadores e construir executáveis em combinações variáveis. Ele também suporta gerações ordenadas (in-*place*) e desordenadas (*out-of-place*) e pode, portanto, suportar múltiplas gerações de uma única árvore fonte. Sendo escrito nas linguagens C, C++ e *Python* há mais ou menos 22 anos atrás. Esse sistema [multiplataforma](https://pt.wikipedia.org/wiki/Multiplataforma) feito para realizar a [geração automatizada](https://pt.wikipedia.org/wiki/Automa%C3%A7%C3%A3o_de_compila%C3%A7%C3%A3o) é um conjunto de aplicações de alto nível e separado do sistema *make*, comum no desenvolvimento em *Unix*.

# CVzone

Este é um pacote de visão computacional que facilita a execução de funções de processamento de imagem e *IA*. No núcleo, ele usa bibliotecas [*OpenCV*](https://github.com/opencv/opencv) e [*Mediapipe*](https://github.com/google/mediapipe) .

# Numpy

O sistema descrito neste artigo importa as bibliotecas do *OpenCV* para processos de reconhecimento facial, o *Numpy* (*Numeric Python*) é usado para realizar cálculos em *Arrays* multidimensionais, sendo *Arrays* de valores numéricos da face de um indivíduo contido em nosso *haar-cascade*, totalmente compilado na linguagem da máquina que faz essa comparação.

# Dlib

É uma biblioteca de *machine learning*, escrita na linguagem *C++*, que implementa diversos algoritmos e ferramentas para suportar problemas dessa área, dentre elas, o reconhecimento facial, que atualmente é uma das mais eficientes para a visão computacional e resolução de problemas de aprendizagem de máquina (Mattos, 2017).

Nos deparamos com o conceito *Haar Cascade* que é um algoritmo de aprendizagem muito promissor, que explicamos abaixo:

# Haar Cascade

*Cascade* (cascata, em português) ou *Haar* *Cascade* é o "apelido" de um famoso algoritmo de busca de objetos em imagens, cujo nome oficial é Algoritmo de Viola-Jones (devido aos nomes dos autores). Ele pode ser encontrado na biblioteca *OpenCV*, é um algoritmo de aprendizado baseado em *AdaBoost*, que seleciona um pequeno número de características visuais críticas de uma determinada imagem e as utiliza para a etapa de detecção e classificação de objetos.

A última etapa e não menos importante foi combinar a linguagem de programação Python com as bibliotecas, mas foi aqui também que vimos a necessidade de comprarmos uma *WebCam* com a resolução 1080p a 30 fps.

# Webcam

A Câmera Web é uma [câmera de vídeo](https://pt.wikipedia.org/wiki/C%C3%A2mera_de_v%C3%ADdeo) de baixo custo que capta imagens e que as transfere para um [computador](https://pt.wikipedia.org/wiki/Computador). Pode ser usada para [videoconferência](https://pt.wikipedia.org/wiki/Videoconfer%C3%AAncia), monitoramento de ambientes, produção de vídeo e imagens para edição, entre outras aplicações. Atualmente existem webcams de baixa ou de alta resolução (acima de 2.0 megapixels) e com ou sem [microfones](https://pt.wikipedia.org/wiki/Microfone) acoplados. Algumas webcams vêm com [LED](https://pt.wikipedia.org/wiki/Diodo_emissor_de_luz) (diodos emissores de luz), que iluminam o ambiente quando há pouca ou nenhuma luz externa.

A maioria das webcams é ligada ao computador por conexões [USB](https://pt.wikipedia.org/wiki/USB) e a captura de imagem é realizada por um componente eletrônico denominado [CCD](https://pt.wikipedia.org/wiki/CCD).

Para entendermos mais sobre o projeto foi levantado pesquisas sobre o seu tema central, além de relatarem o que poderia ser integrado a ele e como pode ser utilizado para benefício da sociedade.

# Reconhecimento Facial

Esse sistema de segurança biométrica já existe há mais de 50 anos. Entre 1964 e 1966, uma equipe de pesquisa liderada por Woodrow W Bledsoe realizou experimentos para verificar se "computadores de programação" podiam reconhecer rostos humanos. A equipe usou um scanner rudimentar para mapear a localização da linha do cabelo, olhos e nariz da pessoa. A tarefa do computador era encontrar os seus matches. O computador foi mal-sucedido, segundo Bledsoe, "O problema do reconhecimento facial é dificultado pela grande variabilidade na rotação e inclinação da cabeça, intensidade e ângulo da iluminação, expressão facial, envelhecimento, etc.". Porém, graças ao avanço da tecnologia, atualmente é possível utilizar esse sistema de forma descomplicada.  
Os sistemas de reconhecimento facial podem ser usados para identificar pessoas em fotos, vídeos ou em tempo real.   
Sistemas de tecnologia facial podem variar, mas geralmente tendem a funcionar da seguinte forma:

Etapa 1 - Detecção de rosto: A câmera detecta e localiza a imagem de um rosto, seja sozinho ou em uma multidão. A imagem pode mostrar a pessoa olhando para frente ou de perfil.  
Etapa 2 - Análise de rosto: A maioria dos sistemas usa a tecnologia de câmera 2D, que cria uma imagem plana de um rosto e mapeia "pontos nodais" (tamanho/formato dos olhos, nariz, maçãs do rosto etc.).   
Etapa 3 - Convertendo a imagem em dados: Nessa etapa, o sistema calcula a posição relativa dos pontos e converte os dados em um código numérico (chamado faceprint).

O sistema de reconhecimento facial, assim como outras ferramentas, contém suas vantagens e desvantagens. Dentre as vantagens, podemos destacar maior segurança, crime reduzido, maior comodidade, processamento mais rápido (o processo de reconhecimento de um rosto leva poucos segundos, o que traz benefícios para as empresas que usam o reconhecimento facial) e integração com outras tecnologias, limitando assim a quantidade de investimento adicional necessário para implementá-lo.  
As desvantagens estão mais relacionadas com a questão da privacidade. Alguns temem que o uso de reconhecimento facial juntamente com câmeras de vídeo, inteligência artificial e análise de dados crie o potencial de vigilância em massa, isso poderia restringir a liberdade individual e promover a violação de privacidade. Outro ponto está nos erros que podem ser cometidos pelo sistema, por exemplo identificar uma pessoa falsamente, podendo esta sofrer consequências injustamente. Entretanto, apesar dos riscos, a tecnologia biométrica oferece soluções de segurança atrativas, os sistemas são convenientes e difíceis de duplicar. Esses sistemas continuarão a se desenvolver no futuro - o desafio será maximizar seus benefícios e minimizar seus riscos.

**CONCLUSÃO**

Para a venda desse produto teríamos que incrementar mais funcionalidades ao sistema, tais como verificação de usuários em alta escala, implementação de um melhor banco de dados, seria necessário também um sistema de cadastro para novos rostos, uma câmera com maior desempenho, soluções que se adaptem e adequa mais às necessidades do cliente, com algumas implementações como a do *Raspberry pi* (microprocessador dedicado para testes Machlearning), que rodaria o sistema com uma maior fluidez e desempenho em diferentes ambientes, deixando-o com uma estética mais comercializável e com um excelente custo benefício.

# ABSTRACT

The objective of the work presented here was to develop a tool that captures faces with the use of a webcam and then analyze them and check if they are registered in our database or not. Several concepts were used, such as python programming language, libraries of this language, and other analyzed tools to make our project run as efficiently as possible. A collection of information was necessary in order to understand the origins of this theme and how other tools with facial recognition functionalities are used. All this research made possible the creation and the operation of the project that focuses on security.

**Keywords:** Recognition, Faces, Python, Libraries, Webcam.

# REFERÊNCIAS

**ANTONELLO, R. Livro: Introdução à Visão Computacional com Python e OpenCV**, 2017.  Disponível em: <<https://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2017/02/Livro-Introdu%C3%A7%C3%A3o-a-Vis%C3%A3o-Computacional-com-Python-e-OpenCV-3.pdf/>> Acesso em: 26 dez. 2021.

**Kaspersky: O que é reconhecimento facial – definição e explicação**, 2021.  Disponível em: <https://www.kaspersky.com.br/resource-center/definitions/what-is-facial-recognition/> Acesso em: 24 maio. 2022.

**LABAKI, J.** **Introdução a Python - Módulo A**. Disponível em: <<https://dcc.ufrj.br/~fabiom/mab225/pythonbasico.pdf/>> Acesso em: 13 jan. 2022.

**LEITE, L.; ANTONELLO, R. SECITEC: Versão Final do Artigo**, 2017.  Disponível em: <<https://secitec.luzerna.ifc.edu.br/wp-content/uploads/sites/30/2018/04/Versao-Final-Artigo-para-SECITEC-2017-10-paginas.pdf/>> Acesso em: 13 dez. 2021.

**OPENCV**. Disponível em: <<https://opencv.org/about/>> Acesso em: 14 jan. 2022.

**PYTHON**. Disponível em: <<https://www.python.org/>> Acesso em: 14 jan. 2022.

**REZAEI, M. Creating a Cascade of Haar-Like Classifiers: Step by Step**. Disponível em: <<https://www.cs.auckland.ac.nz/~m.rezaei/Tutorials/Creating_a_Cascade_of_Haar-Like_Classifiers_Step_by_Step.pdf/>> Acesso em: 17 jan. 2022.

**Sistema de reconhecimento facial**. Disponível em: <<https://pt.wikipedia.org/wiki/Sistema_de_reconhecimento_facial/>> Acesso em: 06 mai. 2022.

**Thales Group: Reconhecimento facial - como a tecnologia se expandiu**. Disponível em: <https://www.thalesgroup.com/pt-pt/countries/americas/thales-brazil/dis/governo/inspire-se/reconhecimento-facial/surgimento-da-tecnologia/> Acesso em: 23 mai. 2022.

**UNESC**. Disponível em: <<http://repositorio.unesc.net/handle/1/8128>/> Acesso em: 25 mar. 2022

**Wikipédia**. Disponível em: <<https://pt.wikipedia.org/wiki/Python>/> Acesso em: 18 abr. 2022