**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS**

**FATEC PROFESSOR Jessen Vidal**

**LEONARDO ALEIXO DA SILVA**

**SISTEMA DE IDENTIFICAÇÃO HUMANA EM CÂMERAS DE VIGILÂNCIA**

São José dos Campos

2022

**LEONARDO ALEIXO DA SILVA**

**SISTEMA DE IDENTIFICAÇÃO HUMANA EM CÂMERAS DE VIGILÂNCIA**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Banco de Dados.

**Orientador: Me. Carlos Augusto Lombardi Garcia**

São José dos Campos

2022

**Dados Internacionais de Catalogação-na-Publicação (CIP)**

**Divisão de Informação e Documentação**

SILVA, Leonardo Aleixo

Sistema de Identificação Humana em Câmeras de Vigilância.

São José dos Campos, 2022.

23f.

Trabalho de Graduação – Curso de Tecnologia em Banco de Dados.

FATEC de São José dos Campos: Professor Jessen Vidal, 2022.

Orientador Interno ou Principal: Me, Carlos Augusto Lombardi Garcia.

1. Inteligência Artificial 1. 2. YOLO 2. 3. Câmera de Segurança 3. I. Faculdade de Tecnologia. FATEC de São José dos Campos: Professor Jessen Vidal. Divisão de Informação e Documentação. II. Título

**REFERÊNCIA BIBLIOGRÁFICA**

SILVA, Leonardo Aleixo. **Sistema de Identificação Humana em Câmeras de Vigilância.** 2022. 23f. Trabalho de Graduação - FATEC de São José dos Campos: Professor Jessen Vidal.

**CESSÃO DE DIREITOS**

NOME(S) DO(S) AUTOR(ES): Leonardo Aleixo da Silva

TÍTULO DO TRABALHO: Sistema de Identificação Humana em Câmeras de Vigilância

TIPO DO TRABALHO/ANO: Trabalho de Graduação/2022.

É concedida à FATEC de São José dos Campos: Professor Jessen Vidal permissão para reproduzir cópias deste Trabalho e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Leonardo Aleixo da Silva  Avenida Pedro Friggi, 3100 – Bloco 12 Apto 34  12223-430, São José dos Campos – São Paulo |  |

**LEONARDO ALEIXO DA SILVA**

**SISTEMA DE IDENTIFICAÇÃO HUMANA EM CÂMERAS DE VIGILÂNCIA**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Banco de Dados.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Me, Giuliano Araujo Bertoti**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Me, Eduardo Sakaue**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Me, Carlos Augusto Lombardi Garcia**

**\_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_**

**DATA DA APROVAÇÃO**

**AGRADECIMENTOS**

Agradeço ao professor e orientador Carlos Augusto Lombardi Garcia pelo apoio no desenvolvimento e instrução de tecnologias corretas, aos demais professores pelos conhecimentos transmitidos e aos colegas de faculdade e trabalho que incentivaram o desenvolvimento.

**RESUMO**

Neste trabalho foi desenvolvido duas aplicações que realizam a gestão e processamento das imagens recebidas por câmeras de segurança para identificação humana quando há movimentos. Essas câmeras são configuradas para enviar dados a um servidor FTP onde é armazenado e processado os frames, a aplicação que faz a identificação faz todo tratamento de dados de imagens, como clareamento, modificação de formato e validação de binários para que a segunda aplicação onde será distribuído aos serviço de atendimento mais próximo e a central de segurança, utilizando como principal ferramenta a linguagem de programação Python e rede neural convolucional YOLOV4. A aplicação apresentou os resultados esperados de processamento requisitado após o uso dos drivers de processamento em GPU CUDA, com isto, é possível paralelizar os processamentos de imagens entre CPU e GPU, mais testes estão sendo executados em ambiente de homologação para garantir a qualidade de assertividade na identificação humana nas imagens.

**Palavras-Chave**: CPU; GPU; Python; YOLOV4.

**ABSTRACT**

In this paper two program was developed that manage and process the images received from security cameras when have movement. The cameras are configured to send data to the FTP server where is stored and processed the frames, the application that identify the humans, does the treatment of images like, increase the brightness, modify format, and validate binaries, with this the human detection will have more accuracy. After this process, the second application is notified where it will be delivered to the security service configured, and to make all processing Python was used as main program language tool and neural network YOLOV4. The application gets great results as expected using the drivers to process in GPU CUDA, this makes possible to parallelize the processing between CPU and GPU, more tests will be done on homologation environment to assert quality in human identifications.

**Keywords**: CPU; GPU; Python; YOLOV4.

**LISTA DE FIGURAS**

[Figura 1 – Rede Neural Convolucional DeepLearningBook, 2021). 15](#_Toc107418758)

[Figura 2 - Fluxograma de processo de análise. 17](#_Toc107418759)

[Figura 3 – Processo de análise em Python com OpenCV2 19](#_Toc107418760)

[Figura 4 - Resultado obtido após processamento de imagem. 20](#_Toc107418761)

**LISTA DE ABREVIATURAS E SIGLAS**

CNN Convolutional Neural Network

CPU Central Processing Unit

IP Internet Protocol

FTP File Transfer Protocol

**SUMÁRIO**

[1. INTRODUÇÃO 11](#_Toc107418805)

[1.1. Objetivos do Trabalho 11](#_Toc107418806)

[1.2. Conteúdo do Trabalho 11](#_Toc107418807)

[2. FUNDAMENTAÇÃO TÉCNICA 13](#_Toc107418808)

[2.1. Inteligência artificial 13](#_Toc107418809)

[2.2. Rede neural 13](#_Toc107418810)

[2.3 Machine Learning 13](#_Toc107418811)

[2.4 Deep Learning 13](#_Toc107418812)

[2.5 Git 14](#_Toc107418813)

[2.6 Typescript 14](#_Toc107418814)

[2.7 Python 14](#_Toc107418815)

[2.8 Java 14](#_Toc107418816)

[2.9 YOLO 14](#_Toc107418817)

[2.10 Spring 14](#_Toc107418818)

[2.11 Angular 2 14](#_Toc107418819)

[2.12 MySql 15](#_Toc107418820)

[2.13 OpenCV 15](#_Toc107418821)

[2.14 Convolutional Neural Network (CNN) 15](#_Toc107418822)

[3. DESENVOLVIMENTO 16](#_Toc107418823)

[3.1. Requisitos 16](#_Toc107418824)

[3.2. Requisitos Funcionais 16](#_Toc107418825)

[3.3 Requisitos Não Funcionais 16](#_Toc107418826)

[3.4 Arquitetura do Sistema 17](#_Toc107418827)

[3.5 FTP 17](#_Toc107418828)

[3.6 Análise da imagem 17](#_Toc107418829)

[3.7 Processamento final 19](#_Toc107418830)

[4. RESULTADOS 21](#_Toc107418831)

[4.1. Resultados Técnicos 21](#_Toc107418832)

[4.2. Aprendizagens 21](#_Toc107418833)

[5. CONSIDERAÇÕES FINAIS 22](#_Toc107418834)

[5.1. Trabalho Futuros 22](#_Toc107418835)

[REFERÊNCIAS 23](#_Toc107418836)

# 1. INTRODUÇÃO

Através da necessidade de atendimento ágil por parte da segurança ao público identificou-se uma oportunidade de utilizar as tecnologias de inteligência artificial para acelerar o processo de identificação e notificação dos agentes responsáveis pelo atendimento.

Com base nisso foi elaborado um sistema capaz de gerenciar câmeras de vigilância com eventos de movimento, processar as imagens da mesma e notificar o responsável da câmera.

Muitas pessoas estão familiarizadas com o reconhecimento facial por usar a tecnologia de face ID para desbloqueio do celular. Essa tecnologia não é apenas um banco de dados contendo uma quantidade enorme de dados para validar o rosto e sim uma matemática capaz de calcular através dos traços do rosto suas características e distinguir se as informações são favoráveis ou não.

Essa tecnologia com um processamento de dados avançado possui outras funções como identificação de pessoas em câmera de segurança, identificação de objetos, cálculos ágeis entre outros.

## 1.1. Objetivos do Trabalho

O objetivo geral deste trabalho é desenvolver dois sistemas; O primeiro sendo capaz de configurar os ambientes de transferência de imagens e relatórios de capturas; O segundo seria capaz de identificar seres humanos em intervalos de tempo configurados no primeiro sistema.

Para a consecução deste objetivo foram estabelecidos os objetivos específicos:

* + Identificação das melhores tecnologias de detecção humana no mercado.
  + Criação de um gerenciamento de servidor FTP.
  + Criação de alerta por detecção humana.
  + Criação de um dashboard para detecção.

## 1.2. Conteúdo do Trabalho

O presente trabalho está estruturado em seis Capítulos, cujo conteúdo é sucintamente apresentado a seguir:

**Capítulo 1** Apresenta informações de introdução ao trabalho.

**Capítulo 2** Apresenta a fundamentação das tecnologias.

**Capítulo 3** Apresenta o desenvolvimento da solução.

**Capítulo 4** Apresenta os resultados obtidos e aprendizagens.

**Capítulo 5** Apresenta as considerações finais deste trabalho a partir da análise dos resultados obtidos.

**Capítulo 6** Apresenta as referências utilizadas para desenvolvimento.

# 2. FUNDAMENTAÇÃO TÉCNICA

Este capítulo tem por objetivo apresentar as tecnologias e termos utilizados para o entendimento da identificação humana por imagem.

O capítulo é subdivido em seções que descrevem o conceito de inteligência artificial.

Seções posteriores mostram uma breve explicação sobre as tecnologias, Sistema de controle de versão de dados (Github), Linguagens de programação Typescript, Java e Python. Os frameworks e bibliotecas YOLO, CNN (Convolutional Neural Network), Spring e Angular 2. Banco de dados utilizado Mysql.

## 2.1. Inteligência artificial

Esta tecnologia está relacionada ao processo e a capacidade de pensamento superpoderoso e a análise de dados para uma finalidade em específico, por exemplo, identificação de mudanças climáticas em imagens meteorológicas, projeções de lucros futuros, identificação de objetos entre outros (ORACLE, 2022).

## 2.2. Rede neural

Redes neurais são processamentos que refletem o pensamento humano, permitindo que programas de computador reconheçam padrões e resolvam problemas comuns nos campos de Inteligência Artificial (IA), usando Machine Learning e Deep Learning (IBM, 2022).

## 2.3 Machine Learning

É uma tecnologia onde os computadores têm a capacidade de aprender de acordo com as respostas esperadas por meio associações de diferentes dados, os quais podem ser imagens, números e tudo que essa tecnologia possa identificar. Machine Learning é o termo em inglês para a tecnologia conhecida no Brasil como aprendizado de máquina (IBM, 2022).

## 2.4 Deep Learning

É um tipo de machine learning que treina computadores para realizar tarefas como seres humanos, o que inclui reconhecimento de fala, identificação de imagem e previsões. Em vez de organizar os dados para serem executados através de equações predefinidas, o deep learning configura parâmetros básicos sobre os dados e treina o computador para aprender sozinho através do reconhecimento padrões em várias camadas de processamento (SAS, 2022)

## 2.5 Git

É um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo (ATLASSIAN, 2022).

## 2.6 Typescript

É uma linguagem de programação de código aberto desenvolvida pela Microsoft. É um superconjunto sintático estrito de JavaScript e adiciona tipagem estática opcional à linguagem (MICROSOFT, 2022).

## 2.7 Python

É uma,linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte (PYNTHON SOFTWARE FOUNDATION, 1991).

## 2.8 Java

Java é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems (SUN MICROSYSTEMS, 1991).

## 2.9 YOLO

YOLO (You Only Look Once) é um método de detecção de objetos de passada única que utiliza uma rede neural convolucional como extrator de características (JOSEPH REDMON, 2015).

## 2.10 Spring

O Spring é um framework Java criado com o objetivo de facilitar o desenvolvimento de aplicações, explorando, para isso, os conceitos de Inversão de Controle e Injeção de Dependências (DEVMEDIA, 2022).

## 2.11 Angular 2

É uma plataforma de desenvolvimento mobile e web desktop, baseada em TypeScript (GOOGLE, 2016)

## 2.12 MySql

O MySQL é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface (ORACLE, 1995).

## 2.13 OpenCV

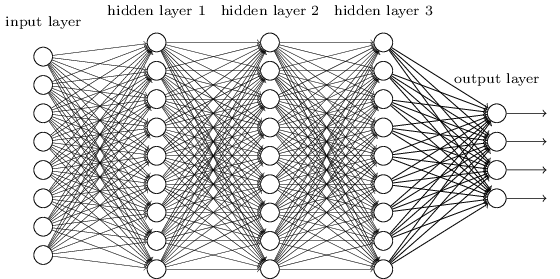
É uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de Visão computacional, bastando seguir o modelo de licença BSD Intel (INTEL, 2000).

## 2.14 Convolutional Neural Network (CNN)

Uma Rede Neural Convolucional (ConvNet / Convolutional Neural Network / CNN) é um algoritmo de Aprendizado Profundo que pode captar uma imagem de entrada, atribuir importância (pesos e vieses que podem ser aprendidos) a vários aspectos / objetos da imagem e ser capaz de diferenciar um do outro.

A Figura 1 representa como funciona uma rede neural, onde se recebe uma entrada e vai passando por filtros distribuindo para outros filtros da imagem até chegar em um resultado. É como se estivesse resolvendo uma equação matemática seguindo as regras de ordem de cálculo.

Figura 1 – Rede Neural Convolucional DeepLearningBook, 2021).



Fonte: Deep Learning Book, (2021)

# 3. DESENVOLVIMENTO

Neste capítulo será abordado o desenvolvimento do projeto. A primeira seção abordará a arquitetura, assim como os requisitos do projeto. A segunda seção será voltada às tecnologias e ferramentas utilizadas para tornar o projeto possível.

## 3.1. Requisitos

Para que o sistema seja considerado pronto, é necessário atingir alguns requisitos.

## 3.2. Requisitos Funcionais

1. O sistema deve ser capaz de detectar um corpo humano através de uma leitura de imagem em diretório dinâmico.
2. O sistema deve enviar informações ao aplicativo de conversas telegram, a porcentagem de precisão detectada e as pessoas detectadas marcadas com um retângulo azul.
3. O sistema deve ser capaz de gerenciar as pastas onde será armazenado as imagens recebidas.
4. O sistema deverá autenticar um usuário para gerenciamento de módulos e permissões internas.
5. O sistema deverá ser capaz de criar usuários a partir de um usuário administrador geral.
6. O sistema deverá excluir as imagens que não detectar nenhum corpo humano.
7. O sistema deverá manter em uma pasta configurada as imagens processadas e depois do envio apagá-las.
8. O sistema deverá manter em uma pasta configurada as imagens processadas e depois do envio apagá-las.

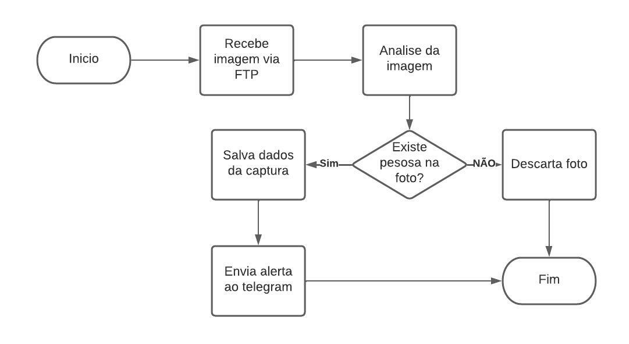
## 3.3 Requisitos Não Funcionais

1. Portabilidade: O sistema deverá ser compilado e executado em qualquer plataforma.
2. O sistema deverá analisar quinze imagens por segundo.
3. O sistema deverá disponibilizar dados privados aos usuários.
4. O sistema deverá ter alta disponibilidade, cerca de 99% do tempo.

## 3.4 Arquitetura do Sistema

A Figura 2 mostra o fluxograma geral de como deverá funcionar o programa. O sistema, por sua vez, é encarregado de criar uma conexão FTP com as câmeras IP’s, e quando detectar um humano em uma imagem recebida e analisada, deverá ser enviado para o aplicativo telegram como alerta a foto e de qual câmera o dado se refere, caso não seja detectado, deverá descartar a imagem.

Figura 2 - Fluxograma de processo de análise.



Fonte: Autor (2022)

## 3.5 FTP

O sistema utiliza uma conexão FTP feita através do Filezila disponibilizando uma pasta de acesso.

## 3.6 Análise da imagem

A análise necessita de duas tecnologias para ser completada, a primeira será o método de detecção YOLO, ele é uma metodologia de IA convolucional que possui uma base de dados, a segunda será o OpenCV que irá utilizar esse método YOLO para fazer a detecção. O motivo do uso de OpenCV é devido a sua alta otimização em processadores INTEL e integração com CUDA (API de conexão com placas de vídeo NVIDIA).

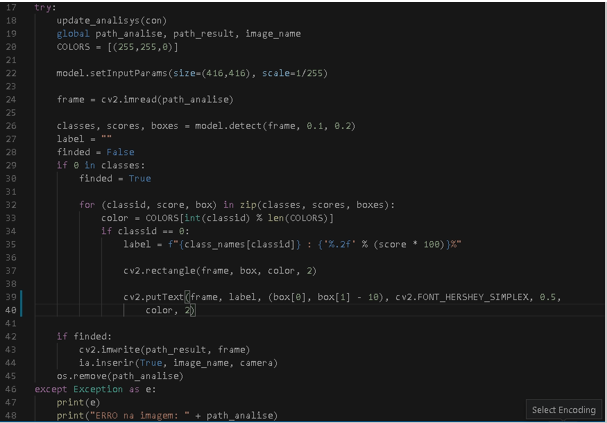
A biblioteca OpenCV possui diversos algoritmos em C++ que viabilizam (OPENCV, 2013):

* Captura de imagens: é possível através da OpenCV acessar câmeras embutidas, USB ou até Câmeras IP, e através destas obter imagens (também chamado de quadros);
* Modificação e pré-processamento de imagens;
* Detecção de objetos: para o caso de faces humanas, é possível detectar a face frontal e perfis esquerdo e direito;
* Reconhecimento facial: a OpenCV possui uma classe dedicada a esta atividade, denominada FaceRecognizer, que faz a previsão de uma determinada face baseada em imagens armazenadas em banco de dados.

A Figura 3 mostra como é feito o processo de análise da imagem em linguagem de programação. O método recebe os parâmetros necessários para trabalhar com a imagem:

* Linha 20 - Define qual será a cor utilizada no retângulo de marcação quando detectar uma pessoa.
* Linha 22 - Representa a configuração dos parâmetros de escala da imagem, como por exemplo, altura e comprimento.
* Linha 24 - Transforma a imagem em uma variável.
* Linha 26 – Faz a detecção baseada no modelo definido (YOLO).
* Linha 32 – Percorre os dados recuperados da detecção para fazer análise humana.
* Linha 34 – Valida se existe ser humano encontrado na foto
* Linha 35, 37 & 39 – Adicionam a imagem os dados de porcentagem, retângulo e texto descritivo.

Figura 3 – Processo de análise em Python com OpenCV2



Fonte: Autor (2022)

## 3.7 Processamento final

A Figura 4 representa o resultado ao aplicar todo cálculo feito pela inteligência artificial, com a borda em volta do resultado obtido.

**Figura 4 - Resultado obtido após processamento de imagem.**



Fonte: Autor (2022)

# 4. RESULTADOS

Para maior precisão nos resultados é necessário um número significativo de treinamentos para a rede neural, indicando tanto os valores positivos e negativos em relação a aprendizagem.

A Tecnologia YOLOv4 é uma rede bem treinada e disponibilizada gratuitamente para uso a qualquer programador.

Para a área de segurança a tecnologia de identificação humana com redes neurais são altamente recomendáveis.

## 4.1. Resultados Técnicos

O processamento final chegou a ser 15 imagens por segundo em um ambiente com 8 gigabyte de memória ram e um CPU I5 7600.

O processamento final chegou a ser 25 imagens por segundo em um ambiente com 8 gigabyte de memória ram e uma placa de vídeo GFORCE 750 TI.

## 4.2. Aprendizagens

Toda rede neural convolucional deve ser treinada com muitos dados de forma assertiva aos pontos falsos e verdadeiros.

OpenCV é uma ótima biblioteca para utilização das redes neurais YOLO além de fornecer muitas funcionalidades de tratamento de imagens.

# 5. CONSIDERAÇÕES FINAIS

Este capítulo visa apresentar algumas das considerações a respeito do

desenvolvimento do trabalho, experiências, conclusões, assim como sugestões para trabalhos futuros.

## 5.1. Trabalho Futuros

Para trabalhos futuros, sugere-se a otimização do processamento de imagens com implementação de novas redes neurais ou novos treinamentos para a rede neural, como por exemplo YOLOv5.

# REFERÊNCIAS

IBM. **Apresentação e Definição sobre o que são as Redes Neurais.**Disponível emhttps://www.ibm.com/br-pt/cloud/learn/neural-networks#:~:text=As%20redes%20neurais%20refletem%20o,machine%20learning%20e%20deep%20learning.

IBM. **Apresentação e Definição sobre o que é Machine Learning.**  
Disponível em  
https://www.ibm.com/br-pt/analytics/machine-learningSAS. **Apresentação e Definição sobre o que é Deep Learning.**  
Disponível em  
https://www.sas.com/pt\_br/insights/analytics/deep-learning.html

ORACLE. **Apresentação e Definição sobre o que é Inteligência Artificial.**   
Disponível em   
https://www.oracle.com/br/artificial-intelligence/what-is-ai/#:~:text=Em%20termos%20mais%20simples%2C%20a,base%20nas%20informa%C3%A7%C3%B5es%20que%20coletam.

YOLO. **Apresentação e Definição da metodologia YOLO.**  
Disponível em   
https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/