**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.

29f.

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.

**REFERÊNCIA BIBLIOGRÁFICA**

SILVA, Leonardo Aleixo. **Sistema de Identificação Humana em Câmeras de Vigilância.** 2022. 29f. 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, Diogo Branquinho Ramos**

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

**Me, Lucas Gonçalves Nadalete**

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

**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**

Este trabalho foi desenvolvido em parceria com a empresa Grupo Solutec, com objetivo de agilizar os processos de identificação a invasões as instituições públicas até o acionamento do agente de segurança mais próximo para averiguação do ocorrido, essas invasões levaram a roubos de pertences dos locais, como botijões de gás e fios de cobre. Através de câmeras de segurança com detecção de movimento instaladas, foi possível configurar um servidor FTP com intuito de receber as imagens geradas pelo evento de movimento e então após processá-las identificar se foi um movimento humano ou não. Para identificação humana as imagens passam por um processo, em que ela será clareada, convertida para os formatos JPEG ou PNG e validação de binários, e então, será enviada para identificação usando OpenCV e a rede neural YOLOV4. Caso seja identificado, o agente de segurança será notificado com a imagem e endereço da câmera. A aplicação apresentou os resultados esperados de processamentos após o uso de GPU com o driver CUDA da Nvidia, com isto, é possível paralelizar os processamentos de imagens entre o CPU e GPU, chegando a processar até 25 imagens por segundo. Para desenvolvimento dessa aplicação foram utilizadas as tecnologias, Java, Python, MySQL, Angular 2, OpenCV, YOLOV4 e CUDA.

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

**ABSTRACT**

This paper was developed in cooperation with the company Grupo Solutec, the objective was more agile in the process of invasion identifications in public buildings and notify the nearest security agent, in additional to the invasions, there were also stolen such as gas canisters and copper wires. Using the technology movement detection in security cameras installed, it was possible to configure the FTP server to receive the generated imagens from movement events and then process the images to identification, like clarify, convert to JPEG or PNG e binary validations, and then send to identification using OpenCV and neural network YOLOV4. In case of valid identification, the agent will be notified com the image and address. The application gets expected results in processing after use CUDA driver from Nvidia, this driver helps to parallelize the process between CPU and GPU, and it to be capable to process 25 images per second. To development of this application was used the technologies, Java, Python, MySQL, Angular 2, OpenCV, YOLOV4 and CUDA.

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

**LISTA DE FIGURAS**

[Figura 1 - Rede Neural Convolucional DeepLearningBook, 2021) 17](#_Toc111396621)

[Figura 2 - Comunicação backend 18](#_Toc111396622)

[Figura 3 - Comunicação frontend 19](#_Toc111396623)

[Figura 4 - Comunicação de câmeras e servidor 19](#_Toc111396624)

[Figura 5 - Fluxograma processamento de imagem 21](#_Toc111396625)

[Figura 6 - Processo de análise em Python com OpenCV2 23](#_Toc111396626)

[Figura 7 - Captura da rede neural e classificações da YOLO 23](#_Toc111396627)

[Figura 8 - Envio de notificação ao telegram com Java 24](#_Toc111396628)

[Figura 9 - Resultado obtido após processamento de imagem 25](#_Toc111396629)

[Figura 10 - Dashboard aplicação 26](#_Toc111396630)

[Figura 11 - Configuração de câmera 27](#_Toc111396631)

[Figura 12 - Leituras em tempo real 27](#_Toc111396632)

[Figura 13 - Visualização de imagem em tempo real 28](#_Toc111396633)

[Figura 14 - Notificação no Telegram 29](#_Toc111396634)

**LISTA DE ABREVIATURAS E SIGLAS**

CNN Convolutional Neural Network

CPU Central Processing Unit

IP Internet Protocol

FTP File Transfer Protocol

CUDA Compute Unified Device Architecture

YOLO You Only Look Once

GPU Graphics Processing Units

RAM Random Access Memory

SSD Solid State Drive

API Interface de Programação de Aplicações

HTTP Hypertext Transfer Protocol

**SUMÁRIO**

[1. INTRODUÇÃO 12](#_Toc111396635)

[1.1. Objetivos do Trabalho 12](#_Toc111396636)

[1.2. Conteúdo do Trabalho 13](#_Toc111396637)

[2. FUNDAMENTAÇÃO TÉCNICA 14](#_Toc111396638)

[2.1. Inteligência artificial 14](#_Toc111396639)

[2.2. Rede neural 14](#_Toc111396640)

[2.3 Machine Learning 14](#_Toc111396641)

[2.4 Deep Learning 14](#_Toc111396642)

[2.5 Git 15](#_Toc111396643)

[2.6 Typescript 15](#_Toc111396644)

[2.7 Python 15](#_Toc111396645)

[2.8 Java 15](#_Toc111396646)

[2.9 YOLO 15](#_Toc111396647)

[2.10 Spring 15](#_Toc111396648)

[2.11 Angular 2 16](#_Toc111396649)

[2.12 MySQL 16](#_Toc111396650)

[2.13 OpenCV 16](#_Toc111396651)

[2.14 Convolutional Neural Network (CNN) 16](#_Toc111396652)

[3. DESENVOLVIMENTO 18](#_Toc111396653)

[3.1. Arquitetura backend 18](#_Toc111396654)

[3.2. Arquitetura frontend 18](#_Toc111396655)

[3.3. Arquitetura câmera servidor 19](#_Toc111396656)

[3.4. Requisitos 20](#_Toc111396657)

[3.5. Requisitos Funcionais 20](#_Toc111396658)

[3.6 Requisitos Não Funcionais 20](#_Toc111396659)

[3.7 Arquitetura do Sistema 20](#_Toc111396660)

[3.8 FTP 21](#_Toc111396661)

[3.9 Análise da imagem 21](#_Toc111396662)

[3.10 Processamento final 24](#_Toc111396663)

[4. RESULTADOS 26](#_Toc111396664)

[4.1. Aplicação de gerenciamento 26](#_Toc111396665)

[4.2. Resultados Técnicos 28](#_Toc111396666)

[4.3. Resultados de requisitos 28](#_Toc111396667)

[4.4. Aprendizagens 30](#_Toc111396668)

[5. CONSIDERAÇÕES FINAIS 31](#_Toc111396669)

[5.1. Trabalho Futuros 31](#_Toc111396670)

[5.2. Outras tecnologias 31](#_Toc111396671)

[5.3. Hardware de implementação 31](#_Toc111396672)

[5.4. Resultados técnicos 32](#_Toc111396673)

[5.5. Resultados pessoais 32](#_Toc111396674)

[5.6. Sistema operacional 32](#_Toc111396675)

[REFERÊNCIAS 33](#_Toc111396676)

# 1. INTRODUÇÃO

A empresa Grupo Solutec trabalha com mais de 1000 câmeras instaladas com apenas a tecnologias de detecção de movimento, e as tecnologias de câmeras de segurança mais atuais, possuem detecções de movimento, calor e humana, porém têm um custo elevado em comparação as comuns, com isto, tornaram-se inviável a substituição de todas as câmeras apenas pela sua função.

Através do problema apresentado, criou-se a necessidade de criação da tecnologia de identificação humana para as câmeras que não possuem essa tecnologia, tendo também outras funções como notificação dos agentes responsáveis.

Uma tecnologia comum de identificações humanas é o face ID, uma tecnologia que se baseia em um banco de dados de imagens para comparar com a face de desbloqueio, porém para validação, essas imagens devem seguir um padrão com margem de erro alta, sendo assim, não poderia vir a ser a tecnologia utilizada.

Em pesquisas sobre inteligência artificial e rede neural, entendeu-se que a rede neural é uma matemática de padrões que pode ser replicado e que a tecnologia OpenCV é capaz de interpretar essas matemáticas e aplicá-las a um cenário específico.

Um exemplo comum para explicar a matemática da rede neural é a identificação de um retângulo, tendo as variáveis, como tamanho das laterais, onde um lado pode ser de 4 centímetros e outro pode ser de 20 centímetros, e existem seus padrões que são os ângulos de 90°, ou seja, mesmo tendo tamanhos diferentes sua matemática padrão é o ângulo interno, e com base nesse entendimento seria necessário buscar tecnologias que fossem capazes de criar matemáticas baseadas em seres humanos. A tecnologia com maior eficiência encontrada foi a YOLOV4, uma rede neural convolucional previamente treinada para identificação de objetos.

## 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 (PYTHON SOFTWARE FOUNDATION, 2022).

## 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 (JAVA, 2022).

## 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).

A arquitetura unificada é extremamente rápida, utilizando um modelo básico do YOLO onde chega a processar imagens em tempo real a 45 frames por segundo. Uma versão menor da rede, Fast YOLO, processa 155 frames por segundo, porém tem maior probabilidade de errar nas detecções (CORNELL UNIVERSITY, 2016).

## 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 (ANGULAR, 2016)

## 2.12 MySQL

O MySQL é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface (MYSQL, 2022).

## 2.13 OpenCV

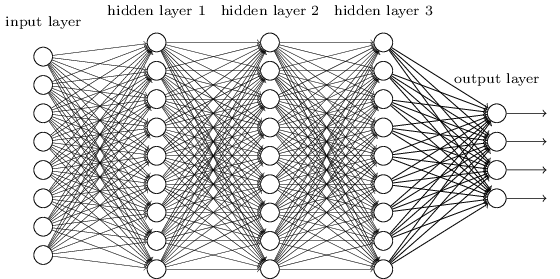
É uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de Visão computacional, bastando seguir o modelo de licença BSD Intel (OPENCV, 2000).

## 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. Arquitetura backend

Os componentes controller e aplicação angular utilizam protocolo http para comunicação, controller e armazenamento de imagens utilizam FTP e model para banco de dados utilizam JDBC, assim como na figura 2.

Figura 2 - Comunicação backend

Diagrama

Descrição gerada automaticamente

Fonte: Autor (2022)

## 3.2. Arquitetura frontend

A figura 3 apresenta a arquitetura do frontend que utiliza a tecnologia node redux para gerenciamento de estado, podendo transferir dados entre componentes de forma mais eficiente e fácil.

Figura 3 - Comunicação frontend

Diagrama

Descrição gerada automaticamenteFonte: Autor (2022)

## 3.3. Arquitetura câmera servidor

As câmeras junto ao servidor se comunicam através dos protocolos de FTP, o Python utiliza comunicação https para armazenar as imagens processadas, representação na figura 4.

Figura 4 - Comunicação de câmeras e servidor

Diagrama

Descrição gerada automaticamente

Fonte: Autor (2022)

## 3.4. Requisitos

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

## 3.5. 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 de identificação deverá funcionar em um sistema operacional Windows

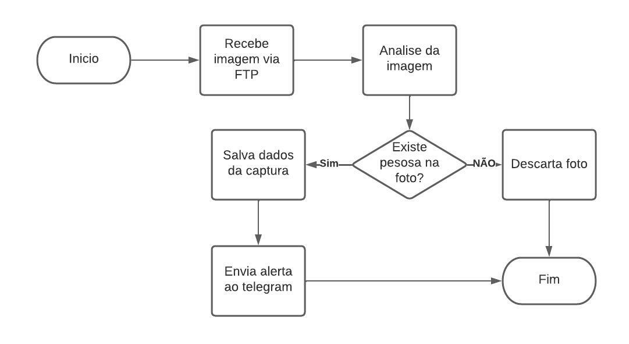
## 3.6 Requisitos Não Funcionais

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.

## 3.7 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 5 - Fluxograma processamento de imagem

Fonte: Autor (2022)

## 3.8 FTP

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

## 3.9 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).

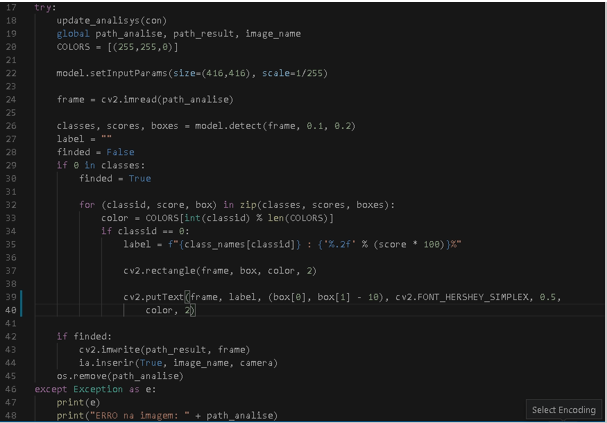
A biblioteca OpenCV possui diversos algoritmos em C++ que viabilizam (OPENCV, 2013):

* 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 6 - Processo de análise em Python com OpenCV2



Fonte: Autor (2022)

Figura 7 - Captura da rede neural e classificações da YOLO

Texto

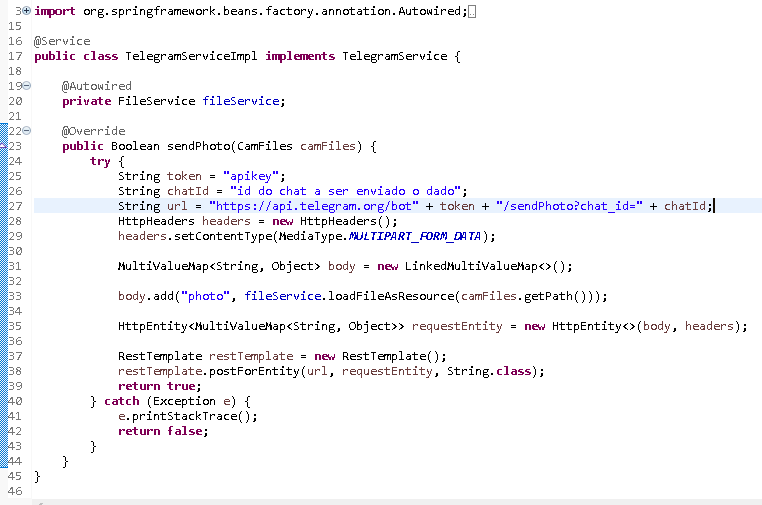
Descrição gerada automaticamente

Fonte: Autor (2022)

A Figura 7 apresenta a captura da rede neural YOLO e as classificações que deverão ser detectadas no processamento das imagens, as variáveis model e class\_names são utilizadas na figura 6 na linha 26.

Os arquivos weights, tiny.cfg e coco.names são arquivos de configuração da rede neural, onde no weights você encontra as regras matemáticas de treinamento da rede neural, no tiny.cfg você encontra as configurações de tamanho da imagem, máximo de cache a ser utilizado, filtros e precisão das imagens, por fim o coco.names é o arquivo de configuração onde você pode apontar quais objetos a rede neural deverá detectar, como por exemplo, humanos, cachorro, gato, moto, placa de trânsito entre outras.

Figura 8 - Envio de notificação ao telegram com Java



Fonte: Autor (2022)

A Figura 8 apresenta a implementação do envio de imagens em Java para a API do Telegram, o método sendPhoto recebe por parâmetro as informações das imagens a ser enviada, contendo o endereço da imagem para ser lida no exemplo acima na linha 33.

Por padrão a API é apenas uma comunicação HTTP, por isso foi utilizado o RestTemplate para envio dessas informações ao telegrama e não uma biblioteca específica para o mesmo.

## 3.10 Processamento final

A Figura 9 representa o resultado ao aplicar todo cálculo feito pela inteligência artificial, com a borda em volta do resultado obtido.

Figura 9 - Resultado obtido após processamento de imagem

Uma imagem contendo edifício, ao ar livre, rua, andando

Descrição gerada automaticamente

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. Aplicação de gerenciamento

A figura 10 apresenta a quantidade de imagens analisadas, total de pessoas detectadas, quantas imagens foram reportadas, a quantidade de imagens processadas por segundo e total de imagens processadas por câmera.

A figura 11 apresenta o modulo de configuração de câmera, onde é possível especificar os horários e dias da semana em que a câmera deverá funcionar.

A figura 12 apresenta a tela de leituras em tempo real, ou seja, quando a imagem for carregada no servidor de armazenamento, ela será carregada nessa tela apresentando seu horário de detecção.

A figura 13 apresenta é uma macro visualização da leitura em tempo real, podendo reportar para o agente de segurança ou validar se a imagem não apresenta problemas.

Figura 10 - Dashboard aplicação

Interface gráfica do usuário, Gráfico

Descrição gerada automaticamente com confiança médiaFonte: Autor (2022)

Figura 11 - Configuração de câmera

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: Autor (2022)

Figura 12 - Leituras em tempo real

Uma imagem contendo foto, diferente, muitos, pessoas

Descrição gerada automaticamente

Fonte: Autor (2022)

Figura 13 - Visualização de imagem em tempo real

Uma imagem contendo andando de, homem, água

Descrição gerada automaticamenteFonte: Autor (2022)

## 4.2. Resultados Técnicos

Para o cálculo de quantidades de imagens processadas foi criado um SLA dentro da aplicação Python, onde se comparou o tempo de início e de término das leituras e a quantidade de imagens analisadas.

Em ambiente de testes foi possível chegar a dois resultados com máquinas diferentes, a primeira máquina com as características de CPU i5 7600 8 gigabytes de memória RAM, armazenamento em SSD e imagens de até 2 megabytes o processamento chegou a 15 imagens por segundo, desde o momento de detecção de imagem até o registro de imagem no banco de dados.

A segunda máquina possui as características CPU i5 7600 8 gigabytes de memória RAM, armazenamento em SSD, imagens de até 3 megabytes, GPU Nvidia GeForce 750 TI o processamento chegou a 25 imagens por segundo.

Com base nisto, conclui-se que a utilização de uma placa de vídeo aumenta significativamente o processamento das imagens e quanto maior o processamento da placa de vídeo, melhor será o desempenho.

## 4.3. Resultados de requisitos

O objetivo de leitura em diretórios dinâmicos e detecção humana foi desenvolvida com a configuração de leitura no sistema de gerenciamento e aplicação de detecção em Python utilizando YOLOV4.

A figura 12 apresenta a notificação no Telegram, apresentando os dados da câmera e a área de detecção humana na imagem.

Figura 14 - Notificação no Telegram

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Autor (2022)

A configuração de pastas de recebimento e armazenamento de imagens foi definida no sistema de gerenciamento.

Os usuários podem ser criados por um administrador através da aplicação de gerenciamento.

As aplicações de software possuem uma característica comum quando uma função do código não está bem configurada ou possui erros, tendo como resultado o término da aplicação, uma vez que esse sistema poderia falhar ao trocar o formato das imagens ou falha de conexão com banco de dados ou até mesmo bloqueios de envio da API do Telegram, então foi implementado uma validação de disponibilidade de serviço no Windows, onde ela a cada 2 minutos validaria se existe um processo Python rodando e para o caso de não estar rodando, iniciaria novamente o sistema, também foi configurado no código uma conversão de dados para apenas os formatos JPEG e PNG, fazendo com que a troca de formatos em que a câmera envia os dados não tenha problema para o sistema, e por último a API do telegrama possibilita 100 envios para cada 30 segundos, para evitar que esse valor seja ultrapassado o código após enviar 29 requisições, espera 3 segundos para enviar as novas requisições.

## 4.4. 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

Visto que a tecnologia YOLOV4 apresentou os resultados esperados em relação a processamento e precisão, decidiu-se implementar futuramente a utilização da nova tecnologia YOLOV5.

A Tecnologia YOLOV5 possui um treinamento e arquitetura mais robusta e complexa em relação a YOLOV4, por isso ela é capaz de aumentar sua precisão e velocidade de processamento, porém sua implementação depende da implementação conjunta de um sistema capaz de validar e alimentar a rede neural com novos resultados para que ela possa identificar seu desempenho e aplicar em cenário real.

## 5.2. Outras tecnologias

A tecnologia YOLOV4 não foi a primeira tecnologia a ser utilizada nas implementações, durante as pesquisas de início a validação humana apenas com classificação em cascata da própria OpenCV resultou em precisões acima de 80%, porém em cenários específicos como chuva e velocidade não apresentou ótimos resultados, chegando a apenas 5 imagens por segundo e precisões abaixo de 40% em imagens com chuva ou de baixa qualidade.

A google possui uma tecnologia chamada Google Lens, que trabalha na identificação de objetos baseados em frames, porém essa tecnologia por uso em API depende do uso de internet, então concluiu-se que não seria uma boa tecnologia a ser utilizado, uma vez em que a latência da internet entre o servidor de hospedagem de imagem e o servidor da google pudesse ser lenta em certos momentos.

## 5.3. Hardware de implementação

A empresa Nvidia desenvolveu hardwares capazes processar imagens em velocidades altíssimas, esse hardware sendo o Jetson Xaview NX Developer Kit, porém esse hardware tem um custo elevado e limite de processamento o que tornou inviável a compra de múltiplos deles para implementação do mesmo, visto que a quantidade de imagens a ser recebidas poderiam variar entre 1 e 1000 por segundo.

## 5.4. Resultados técnicos

O objetivo final deste trabalho era a implementação de dois sistemas, um deles sendo capaz de processar as imagens, identificar humanos e então enviar notificações configuradas em um segundo sistema, que teria as funções de cadastrar os usuários, câmeras, agentes de notificação, diretórios de armazenamento e apresentação de resultados como dashboard, todas estas necessidades foram atendidas e descritas neste documento, conforme os capítulos acima.

## 5.5. Resultados pessoais

O início do desenvolvimento de ambos os sistemas apresentou dificuldades por não conhecimento das tecnologias em rede neural e não conhecimento pleno nas tecnologias de node redux do angular, no entanto em algumas semanas após as implementações, tornou mais claro o entendimento de como cada um funciona, onde se apresentou desempenho na codificação e melhoria nas buscas por entendimentos necessários, ou seja, a implementação dessas tecnologias trouxe um ótimo conhecimento visto que futuramente a maioria dos serviços que possuem um padrão utilizarão inteligências artificiais.

## 5.6. Sistema operacional

As aplicações de gerenciamento e de identificação não foram configuradas para funcionamento específico a um sistema operacional, ambos podem ser instalados em sistemas Unix, Mac OS ou Windows, porém como implementação final, o sistema de gerenciamento foi implementado em nuvem usando sistema Ubuntu 20.04 LTS 4 gigabytes de memória RAM e 2 núcleos de processamento. A aplicação de identificação humana foi configurada em um sistema operacional Windows pois ele deveria ser o servidor de recebimento de imagens conforme a necessidade de requisitos funcionais.

# REFERÊNCIAS

ANGULAR 2. **Conceitos sobre a tecnologia ANGULAR 2.**  
Disponível em   
https://angular.io/guide/architecture, 2022.

CNN. **Conceitos técnicos sobre Convolutional Neural Network.**  
Disponível em   
https://www.ibm.com/cloud/learn/convolutional-neural-networks, 2022.

DEEP LEARNING. **Apresentação e Definição sobre o que é Deep Learning.**  
Disponível em  
https://www.sas.com/pt\_br/insights/analytics/deep-learning.html, 2022

GIT. **Apresentação e Definição da metodologia GIT.**  
Disponível em   
https://www.atlassian.com/br/git/tutorials/what-is-git, 2022.

IBM. **Apresentação e Definição técnica sobre Redes Neurais.**Disponível emhttps://www.ibm.com/br-pt/cloud/learn/neural-networks, 2022.

IBM. **Apresentação e Definição sobre o que é Machine Learning.**  
Disponível em  
https://www.ibm.com/br-pt/analytics/machine-learning, 2022.

JAVA. **Apresentação sobre o que é a tecnologia JAVA.**  
Disponível em   
https://www.java.com/pt-BR/download/help/whatis\_java.html, 2022. 2022.

MySQL. **Apresentação e Definição da tecnologia MySQL.**  
Disponível em   
https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html, 2022.

OPENCV. **Apresentação da tecnologia OPENCV.**  
Disponível em   
https://opencv.org/about/, 2022.

ORACLE. **Conceito e Definição da tecnologia Inteligência Artificial.**   
Disponível em   
https://www.oracle.com/br/artificial-intelligence/what-is-ai/, 2022.

PYTHON. **Documentação da tecnologia PYTHON.**  
Disponível em   
https://docs.python.org/3/, 2022.

SPRING. **Introdução a tecnologia SPRING FRAMEWORK.**  
Disponível em   
https://www.devmedia.com.br/exemplo/como-comecar-com-spring/73, 2022.

TELEGRAM. **Documentação da API TELEGRAM.**  
Disponível em   
https://core.telegram.org/, 2022.

TYPESCRIPT. **Documentação da tecnologia TYPESCRIPT.**  
Disponível em   
https://www.typescriptlang.org/docs/, 2022.

YOLO. **Desenvolvimento com a tecnologia YOLO.**  
Disponível em   
https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/, 2022.

YOLO. **Apresentação e Definição da tecnologia YOLO.**  
Disponível em   
https://www.java.com/pt-BR/download/help/whatis\_java.html, 2022.

YOLO. **Review de algoritmo YOLO.**  
Disponível em   
https://www.sciencedirect.com/science/article/pii/S1877050922001363, 2022.