**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) 16](#_Toc111301688)

[Figura 2 - Comunicação backend 17](#_Toc111301689)

[Figura 3 - Comunicação frontend 18](#_Toc111301690)

[Figura 4 - Comunicação de câmeras e servidor 18](#_Toc111301691)

[Figura 5 - Fluxograma processamento de imagem 20](#_Toc111301692)

[Figura 6 - Processo de análise em Python com OpenCV2 22](#_Toc111301693)

[Figura 7 - Resultado obtido após processamento de imagem 23](#_Toc111301694)

[Figura 8 - Dashboard aplicação 24](#_Toc111301695)

[Figura 9 - Configuração de câmera 25](#_Toc111301696)

[Figura 10 - Leituras em tempo real 25](#_Toc111301697)

[Figura 11 - Visualização de imagem em tempo real 26](#_Toc111301698)

[Figura 12 - Notificação no telegram 27](#_Toc111301699)

**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

**SUMÁRIO**

[1. INTRODUÇÃO 12](#_Toc111303261)

[1.1. Objetivos do Trabalho 12](#_Toc111303262)

[1.2. Conteúdo do Trabalho 13](#_Toc111303263)

[2. FUNDAMENTAÇÃO TÉCNICA 14](#_Toc111303264)

[2.1. Inteligência artificial 14](#_Toc111303265)

[2.2. Rede neural 14](#_Toc111303266)

[2.3 Machine Learning 14](#_Toc111303267)

[2.4 Deep Learning 14](#_Toc111303268)

[2.5 Git 15](#_Toc111303269)

[2.6 Typescript 15](#_Toc111303270)

[2.7 Python 15](#_Toc111303271)

[2.8 Java 15](#_Toc111303272)

[2.9 YOLO 15](#_Toc111303273)

[2.10 Spring 15](#_Toc111303274)

[2.11 Angular 2 15](#_Toc111303275)

[2.12 MySQL 16](#_Toc111303276)

[2.13 OpenCV 16](#_Toc111303277)

[2.14 Convolutional Neural Network (CNN) 16](#_Toc111303278)

[3. DESENVOLVIMENTO 17](#_Toc111303279)

[3.1. Arquitetura backend 17](#_Toc111303280)

[3.2. Arquitetura frontend 17](#_Toc111303281)

[3.3. Arquitetura câmera servidor 18](#_Toc111303282)

[3.4. Requisitos 19](#_Toc111303283)

[3.5. Requisitos Funcionais 19](#_Toc111303284)

[3.6 Requisitos Não Funcionais 19](#_Toc111303285)

[3.7 Arquitetura do Sistema 19](#_Toc111303286)

[3.8 FTP 20](#_Toc111303287)

[3.9 Análise da imagem 20](#_Toc111303288)

[3.10 Processamento final 22](#_Toc111303289)

[4. RESULTADOS 24](#_Toc111303290)

[4.1. Aplicação de gerenciamento 24](#_Toc111303291)

[4.2. Resultados Técnicos 26](#_Toc111303292)

[4.3. Resultados de requisitos 26](#_Toc111303293)

[4.4. Aprendizagens 27](#_Toc111303294)

[5. CONSIDERAÇÕES FINAIS 28](#_Toc111303295)

[5.1. Trabalho Futuros 28](#_Toc111303296)

[REFERÊNCIAS 29](#_Toc111303297)

# 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 (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

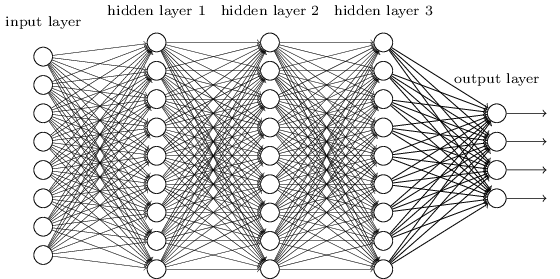
É uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de Visão computacional, bastando seguir o modelo de licença BSD Intel (INTEL, 2000).

## 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. 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.

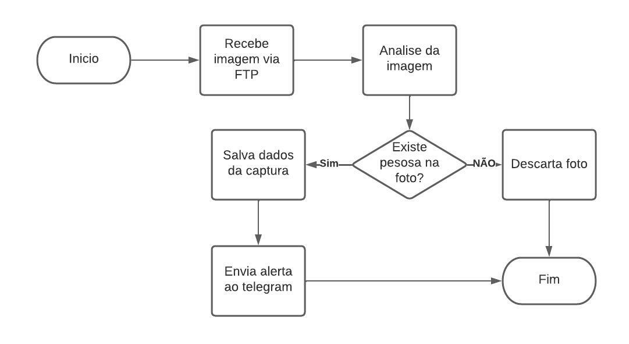
## 3.6 Requisitos Não Funcionais

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).

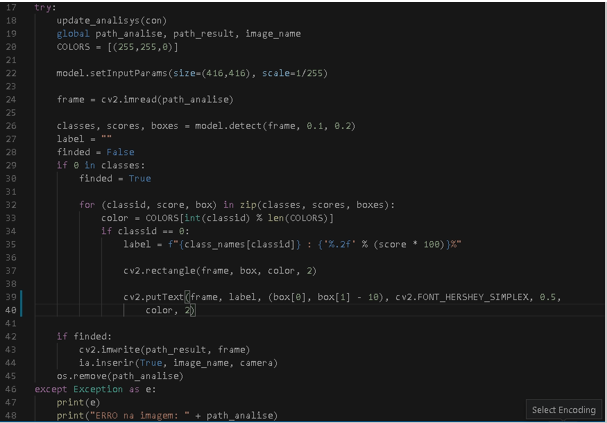
A biblioteca OpenCV possui diversos algoritmos em C++ que viabilizam (OPENCV, 2013):

* 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)

## 3.10 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 7 - 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 8 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 9 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 10 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 11 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 8 - Dashboard aplicação

Interface gráfica do usuário, Gráfico

Descrição gerada automaticamente com confiança médiaFonte: Autor (2022)

Figura 9 - Configuração de câmera

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: Autor (2022)

Figura 10 - Leituras em tempo real

Uma imagem contendo foto, diferente, muitos, pessoas

Descrição gerada automaticamente

Fonte: Autor (2022)

Figura 11 - 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.

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.

## 4.3. 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 12 - 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.

## 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

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/