CONSTRUINDO UMA BASE DE ROTULAGEM DE DADOS POR VISÃO COMPUTACIONAL PARA VALIDAR ASPECTOS DE SEMÂNTICA EM ROBÓTICA ASSISTIVA

Trabalho de Conclusão de Curso

Engenharia da Computação

Richard Jeremias Martins Rocha
Orientador: Carmelo José Albanez Bastos Filho
Coorientador: Antonio Victor Alencar Lundgren

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação

RICHARD JEREMIAS MARTINS ROCHA

CONSTRUINDO UMA BASE DE ROTULAGEM DE DADOS POR VISÃO COMPUTACIONAL PARA VALIDAR ASPECTOS DE SEMÂNTICA EM ROBÓTICA ASSISTIVA

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco da Universidade de Pernambuco.

Orientador: Carmelo José Albanez Bastos Filho
Coorientador: Antonio Victor Alencar Lundgren

Recife
outubro de 2022.

Rocha, Richard Jeremias Martins

Construindo uma base de rotulagem de dados por Visão Computacional para validar aspectos de semântica em Robótica Assistiva / Richard Jeremias Martins Rocha. – Recife - PE, 2022.

xiii, 66 f. : il. ; 29 cm.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) Universidade de Pernambuco, Escola Politécnica de Pernambuco, Recife, 2022.

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho.

Coorientador: Prof. Me. Antonio Victor Alencar Lundgren.

Inclui referências.

1. Análise Semântica. 2. Base de dados. 3. Visão Computacional. 4. Robótica. 5. Tecnologias Assistivas I. Construindo uma base de rotulagem de dados por Visão Computacional para validar aspectos de semântica em Robótica Assistiva . II. Bastos Filho, Carmelo J. A. . III. Universidade de Pernambuco.

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 21/10/2022, às 12h00min, reuniu-se para deliberar sobre a defesa da monografia de conclusão de curso do(a) discente **RICHARD JEREMIAS MARTINS ROCHA**, orientado(a) pelo(a) professor(a) **CARMELO JOSE ALBANEZ BASTOS FILHO**, sob título CONSTRUINDO UMA BASE DE ROTULAGEM DE DADOS POR VISÃO COMPUTACIONAL PARA VALIDAR ASPECTOS DE SEMÂNTICA EM ROBÓTICA ASSISTIVA, a banca composta pelos professores:

SÉRGIO CAMPELLO OLIVEIRA (PRESIDENTE)
CARMELO JOSE ALBANEZ BASTOS FILHO (ORIENTADOR)

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

☒ Aprovada ☐ Aprovada com Restrições* ☐ Reprovada

e foi-lhe atribuída nota: 10,0 (dez)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O(A) discente terá 07 dias para entrega da versão final da monografia a contar da data deste documento.

AVALIADOR 1: Prof (a) **SÉRGIO CAMPELLO OLIVEIRA**

AVALIADOR 2: Prof (a) **CARMELO JOSE ALBANEZ BASTOS FILHO**

AVALIADOR 3: Prof (a)

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

Dedico esse trabalho aos que contribuíram para que eu estivesse aqui hoje.
Sobretudo, aos meus avós Jael Martins e Renilson Rocha.

Agradecimentos

Em primeira instância, gostaria de agradecer a minha mãe, por me formar e ser a pessoa que sou hoje. Meu amor, eu agradeço imensamente por tudo que foi feito por mim até aqui. Estamos mudando o percurso da nossa vida.

Em segundo lugar, gostaria de agradecer aos docentes da Universidade de Pernambuco, a academia de conhecimento na qual percorri um longo caminho durante esses últimos anos. Em especial, gostaria de agradecer aos professores Carmelo José Albanez Bastos Filho e Antonio Victor Alencar Lundgren por me mentorear e aceitar esse desafio de construirmos este trabalho de conclusão de curso.

Em terceira instância, mas nem tão pouco menos importante, gostaria de agradecer aos meus colegas de curso, que foram peças fundamentais para o meu desenvolvimento dentro da Academia. Com o apoio dos mesmos, pude ir cada vez mais longe e, orgulhosamente, hoje estamos aqui.

ESCOLA POLITÉCNICA
DE PERNAMBUCO

Autorização de publicação de PFC

Eu, **Richard Jeremias Martins Rocha** autor(a) do projeto de final de curso intitulado: **CONSTRUINDO UMA BASE DE ROTULAGEM DE DADOS POR VISÃO COMPUTACIONAL PARA VALIDAR ASPECTOS DE SEMÂNTICA EM ROBÓTICA ASSISTIVA;** autorizo a publicação de seu conteúdo na internet nos portais da Escola Politécnica de Pernambuco e Universidade de Pernambuco.

O conteúdo do projeto de final de curso é de responsabilidade do autor.

Richard Jeremias Martins Rocha

Richard Jeremias Martins Rocha

Orientador(a): **Carmelo Jose Albanez Bastos Filho**

Coorientador(a): **Antonio Victor Alencar Lundgren**

Prof, de TCC: **Daniel Augusto Ribeiro Chaves**

Data: 21/10/2022

Resumo

Todo ser humano tem o direito de viver de maneira segura. Com o crescimento da Inteligência Artificial e da Robótica, se torna cada vez mais possível construir e auxiliar o desenvolvimento de tecnologias que busquem melhorar o dia-a-dia da população. Uma análise de imagem a nível semântico promove a extração de suas características de acordo com a percepção humana, permitindo consequentemente a descrição e a organização de seu conteúdo para análise posterior. As informações vitais de uma imagem a partir da semântica são identificadas pelo conteúdo das imagens, principalmente por uma ou mais imagens de objetos com um teor significativo identificável e suas relações mútuas. Ao utilizar a análise semântica, a proposta de verificação de conteúdo de uma imagem permite, em relação a seus conceitos básicos, que a intenção de entrega de conhecimento seja pauta dos trabalhos relacionados a esse tema. A proposta de desenvolvimento de novas bases informativas ajuda o campo da Robótica e da Inteligência Artificial, ao contribuir para a vivência das pessoas que necessitam de Tecnologias Assistivas auxiliando no treinamento e testes de sistemas baseados nessa premissa. Este trabalho tem como objetivo construir uma base de dados que validem semanticamente aspectos de segurança dentro de um cenário comum na vida cotidiana do ser humano. Com a expansão de técnicas de *Deep Learning*, o desenvolvimento de uma base de dados consolidada é, portanto, útil no desenvolvimento de algoritmos que a utilizem em sistemas classificadores inteligentes. A partir das informações obtidas, foi desenvolvido uma base de dados que dispõe de seu conteúdo em dois formatos conhecidos na área da Visão Computacional, onde cada arquivo irá conter a classificação dos objetos encontrados em uma cena como seguro ou inseguro, extraindo dessa forma as características semânticas de uma imagem.

Palavras-chave: Análise Semântica; Base de dados; Visão Computacional; Robótica; Tecnologias Assistivas;

Abstract

Every human being has the right to live safely. With the growth of Artificial Intelligence and Robotics, it becomes increasingly possible to build and help the development of technologies that seek to improve the day-to-day of the population. An image analysis at the semantic level promotes the extraction of its characteristics according to human perception, consequently allowing the description and organization of its content for further analysis. The vital information of an image from semantics is identified by the content of the images, mainly by one or more images of objects with an identifiable significant content and their mutual relationships. By using semantic analysis, the proposal to verify the content of an image allows, in relation to its basic concepts, that the intention of delivering knowledge is the agenda of works related to this theme. The proposal to develop new information bases helps the field of Robotics and Artificial Intelligence, by contributing to the experience of people who need Assistive Technologies, helping in the training and testing of systems based on this premise. This work aims to build a database that semantically validates security aspects within a common scenario in the daily life of human beings. With the expansion of Deep Learning techniques, the development of a consolidated database is, therefore, useful in the development of algorithms that use it in intelligent classification systems. From the information obtained, a database was developed that has its content in two formats known in the area of Computer Vision, where each file will contain the classification of objects found in a scene as safe or unsafe, thus extracting the characteristics semantics of an image.

Keywords: Semantic Analysis; Data base; Computer vision; robotics; Assistive Technologies;

Lista de ilustrações

Lista de tabelas

Lista de símbolos

w	Pesos
x	Vetor de dados
$\hat{x}$	Vetor de predição
θ	Parâmetros ou pesos da rede
$\sum$	Somatório
$*$	Convolução
$\int$	Integral
∞	Infinito
τ	Tau

Lista de abreviaturas e siglas

CNN	Rede Neural Convolucional
DBN	Rede de crenças profundas
DT	Árvore de Decisão
DL	Aprendizagem profunda
GPU	Unidade de Processamento Gráfico
IA	Inteligência Artificial
IHM	Interface Homem-Máquina
ML	Aprendizagem de máquina
MLP	Perceptron Multicamadas
MSE	Erro Quadrático Médio
PDI	Processamento Digital de Imagens
PnP	Perspective-n-Point
ReLU	Rectified Linear Units
RNA	Rede Neural Artificial
RNN	Rede Neural Recorrente
SVM	Support Vector Machine
TA	Tecnologia Assistiva
VC	Visão Computacional
NLP	Processamento de Linguagem Natural

Sumário

1 Introdução

Tecnologia Assistiva é uma designação de termo aos recursos e projetos que contribuem para proporcionar ou auxiliar o desenvolvimento de habilidades funcionais de um ser ou de grupos, e consequentemente promover vida independente e inclusão [1]. Viver no hoje é entender que todas as pessoas têm sua individualidade e formas de lidar com situações adversas no dia-a-dia. Num sentido amplo, é convencional concluir que a evolução das tecnologias hoje em dia busca seguir na direção de tornar a vida mais fácil para seus usuários.

Introduzir o conceito da Tecnologia assistiva mostra que estamos certamente caminhando em busca de uma sociedade mais acolhedora. De acordo com o *American with Disabilities Act* [2], essas tecnologias são "uma ampla gama de equipamentos, serviços, estratégias e práticas concebidas e aplicadas para minorar os problemas funcionais encontrados pelos indivíduos com deficiências".

Essas tecnologias buscam ser conhecidas como um auxílio que promoverá a ampliação de uma habilidade funcional deficitária ou possibilitará a realização da função desejada e que se encontra impedida por circunstância de deficiência [1,3], envelhecimento, ou alguma limitação. Pode-se então dizer que o objetivo desse conceito é proporcionar à qualquer pessoa, sendo comunidades como a de pessoas de maiores ou menores idades e pessoas com deficiência, maior independência, qualidade de vida e inclusão social, através do foco da melhora na acessibilidade, mobilidade, controle de seu ambiente, como também habilidades de seu aprendizado e trabalho.

Visão computacional (VC) é uma área da computação e da Inteligência Artificial (IA) que interpreta e extrai dados relevantes de vídeos e imagens para que decisões possam ser feitas ou permitir que conhecimentos possam ser aprendidos para futuras decisões [4]. A ideia é simular a visão humana, entender distâncias e compreender o que há em um ambiente.

Com o surgimento de algoritmos de Inteligência Artificial, Aprendizagem de máquina e *feature engineering*, permitiu-se o crescimento na construção de sistemas informatizados. Em certo momento, foi percebido que a combinação de redes neurais artificiais e um grande volume de dados, oriundos dos mais diversos aparelhos como dispositivos móveis e dados gerados por usos em redes sociais por exemplo, trazia um enorme ganho de desempenho e

uma possibilidade de estudo com essas tarefas, a área de visão computacional alavancou ainda mais e trouxe diversas aplicações para as tarefas do dia a dia [5].

Com classificação e reconhecimento de padrões em imagens, conseguimos tipificar uma cena utilizando alguns conceitos desse nicho da Computação, facilitando a separação dos dados gerados pelo resultado, e criando um ambiente organizado para agilizar os processos. Com a detecção e comparação dos objetos, conseguimos comprovar que aquele objeto está de fato no ambiente trazendo algum tipo de risco [6] para as pessoas que circulam ou entram em contato com aquele espaço, reduzindo danos e aumentando a segurança dos diversos usuários naquele lugar [7].

A robótica socialmente assistiva (SAR) é o campo que faz uso de robôs para auxiliar humanos em tarefas diversas, comumente voltada à medicina cirúrgica e terapêutica [8] e que vem sido amplamente estudada para diversas finalidades, com avanços voltados à processos cirúrgicos [9], simulação de companheirismo [10], assistência à saúde mental [11] e cuidadores robóticos [12]. Estas aplicações criam não apenas novos mercados para empresas de tecnologia voltadas à medicina e terapia, mas também expandem os horizontes de comodidade e facilidade da vida humana.

Modelos baseados em aprendizagem profunda se mostram extremamente eficientes em tarefas de Visão Computacional, com a utilização de Redes Neurais Artificiais Convolucionais (CNNs). Os desafios de detecção e reconhecimento de textos e objetos em cenas naturais são bem solucionados utilizando estas técnicas, no entanto alguns desafios ainda persistem, em principal a mobilidade destas técnicas no que trata das limitações de hardware das plataformas robóticas, com algumas novas abordagens para o problema de mobilidade sendo recentemente pesquisadas [13].

Análise Semântica (AS) é um campo da aprendizagem de máquina que aborda o contexto envolvendo artefatos de aprendizado, comumente utilizado na área de processamento de linguagem natural [14]. A aprendizagem profunda também beneficiou imensamente a AS e o Processamento de Linguagem Natural (NLP) de maneira geral, explicam o importante papel que as arquiteturas profundas, incluindo o uso de CNNs, possuíram nos avanços da área [15].

Contudo, esse campo de utilização de técnicas de classificação em convergência com algoritmos de aprendizagem de máquina, como a AS, deverá auxiliar no processamento dos dados para gerar abordagens de problemas no campo da Robótica Assistiva. A problemática a ser sanada é a falta de dados diante desse contexto, sendo proposta principal desse projeto

gerar e validar o conteúdo a ser utilizado para análise, promovendo assim resultados que permitam abordagens para análise semântica visual em aplicações robóticas.

Há ainda lacunas que devem ser avaliadas, como a extração automática de informações semânticas importantes sem necessariamente a necessidade de ontologias, ou seja organização de informações que almejam uma representação formal de conhecimento, sendo este o limite considerável para tarefas de visão computacional relacionada à AS, o que é almejado através dos exemplos de interseção de informações desses contextos discutidos no artigo que trata de uma revisão sistemática da AS da VS em Robótica Socialmente Assistiva [16].

1.1 Motivação

O presente trabalho busca contribuir para o uso de algoritmos de inteligência artificial, com ênfase na camada de Visão Computacional, aplicados a área de Robótica,e tem como objetivo incentivar o uso de sistemas classificadores a partir de ferramentas de classificação, com ênfase em facilitar e aprimorar o processo de detecção de imagens quando se trata na criação de uma base de dados. Além disso, busca-se contribuir para a construção de uma ferramenta que irá detectar objetos com altos graus de perigo dentro de um contexto. Essa ferramenta está sendo desenvolvida em paralelo, e tem como objetivo o barateamento do custo associado a sistemas robóticos que auxiliam em tarefas similares, fazendo com que essas ações se tornem mais acessíveis a parcela da população que muitas vezes se encontra impossibilitada de entrar em contato com tecnologias semelhantes. Por consequência, poderá influenciar uma melhor qualidade de vida para as pessoas que se encontram em situações análogas.

1.2 Objetivos

1.2.1 Objetivo geral

O presente trabalho de conclusão de curso propõe o estudo, implementação, e avaliação da construção de uma base para análise semântica de uma cena utilizando Visão Computacional, no contexto de aplicações em robótica assistiva. É idealizado a extração de contexto de objetos em cenas naturais para solução de problemas, como o reconhecimento de objetos periculosos. Mais especificamente, este projeto tem como objetivo propor uma

abordagem que utiliza técnicas de aprendizagem de máquina para detectar e reconhecer regiões de interesse em cenas do cotidiano.

O projeto aqui desenvolvido permitirá que o conteúdo disposto possa ser utilizado para treinar algoritmos que validem os objetos em um ambiente, identificando caso ele seja um dos objetos de estudo escolhidos para ser analisado em um contexto, guiando assim soluções de robótica assistiva, ou seja, para que os robôs possam identificar posteriormente a cena natural disposta.

1.2.2 Objetivos específicos

Para conclusão do objetivo geral, alguns objetivos específicos são estipulados. Nesse sentido, algumas metas específicas foram definidas para definir o escopo do projeto e estruturar seu desenvolvimento. São elas:

- Escolha das variáveis semânticas para o contexto de análise;
- Criar a base a partir de uma ferramenta de classificação;
- Realizar uma primeira validação da base com uma abordagem de aprendizado de máquina;
- Auxiliar o treinamento do algoritmo utilizado nesta validação para reconhecer ativamente os objetos;

1.3 Estrutura do trabalho

No Capítulo 2, 3, 4 são apresentadas revisões a respeito dos temas que este trabalho engloba para que o leitor possa se habituar com o contexto atrelado ao trabalho, expondo um pouco sobre análise semântica, interface homem-máquina, inteligência artificial, redes neurais artificiais, redes neurais convolucionais, a arquitetura desses conceitos, tal qual a modelagem do processamento dos conteúdos a serem detectados. No Capítulo 5, é detalhada toda a metodologia que foi definida para a execução do trabalho, como as ferramentas utilizadas e algumas atividades requeridas para utilização dos dados. No Capítulo 6 são apresentados os resultados gerados a partir da base construída e os resultados encontrados a respeito do tema trabalhado. No Capítulo 7 são apresentadas as conclusões e trabalhos futuros com base em toda construção desenvolvida no presente trabalho.

2. Semântica e Interação Homem-Máquina

Os primeiros detectores de objetos eram baseados em recursos feitos a mão e empregavam uma abordagem baseada em um fator de avaliação que era computacionalmente ineficiente e menos preciso em seu resultado [17]. As técnicas modernas aplicam diferentes conceitos para auxiliar no crescimento da assertividade na detecção de itens em uma imagem.

Existem duas técnicas distintas na segmentação de imagens que ajudam os projetos de visão computacional. São elas:

Segmentação Semântica – Envolve detectar objetos dentro de uma imagem e agrupá-los com base em categorias definidas.

Por exemplo: Em uma cena de rua, se desenharia limites e rotularia itens em subgrupos como humanos, automóveis, bicicletas, semáforos, passarelas, cruzamentos, pistas, etc.

Segmentação de instância – Além do que é descrito na segmentação semântica, envolve também a detecção de objetos dentro de categorias definidas.

Por exemplo: Na mesma cena utilizada acima, seria feito limites individuais para cada uma das categorias e rotularia exclusivamente como humanos, objetos e automóveis.

Embora a rotulagem de segmentação de instância seja custosa no seu processamento computacional, ela é um dos métodos mais robustos e abrangentes de obter a detecção de objetos na análise de imagens [18]. Além disso, o valor informativo daquela imagem é amplificado ao avaliar vídeos analisando quadro-por-quadro de sua composição. Ou seja, é possível extrair conteúdo inteligente quando se identifica exclusivamente cada instância de objetos em uma imagem que é segmentada por categorias definidas.

A segmentação panóptica combina a segmentação de instância e semântica, sendo assim, promove a associação de cada pixel do conteúdo a dois valores: seu rótulo e um número de instância. Ou seja, a segmentação panóptica trabalhará com os seguintes fatores: camadas, instâncias e objetos [17].

Uma implementação comum utilizada dependeria de técnicas de segmentação de imagem desenvolvidas a partir de uma Rede Neural Convolucional que buscaria envolver o desenho a partir de limites em nível de pixel dos objetos em uma imagem, sendo chamados também de bounding box na área de classificação de imagens [19]. Geralmente essas caixas

delimitadoras são descritas da forma de vetor com 4 valores: bx,by,bh,bw. Ou seja, sua posição no espaço da imagem.

2.1 Interface Homem-Máquina

Interface Homem-Máquina (IHM) é uma interface gráfica que permite o homem interagir com uma máquina. Existem vários tipos de IHM no cenário mundial, mas independente de local de aplicação, os seus elementos básicos de uma IHM são:

- Entrada de dados: a IHM recebe do operador instruções de como ele quer que determinada máquina funcione. O operador pode fornecer estes dados de entrada de diversas formas, seja por periféricos simples, como teclados e mouses ou até mesmo por meios mais complexos, como redes sem fio ou similares.
- Saída de dados: forma como a IHM repassa para o operador as informações vindas da máquina, como sinais luminosos (LEDs) ou por meio de interfaces gráficas, como animações, janelas de mensagem ou gráficos. Desta forma, o operador pode interagir com os dados disponíveis sobre o equipamento e ter conhecimento das atividades do mesmo.
- Sinais de entrada de emergência: sinalização de erro ou comandos de emergência que ficam disponíveis para o operador em situações de urgência e exigem um curto tempo de resposta, como alarmes e botões de parada de emergência.

2.1.1 IHM para supervisão

Desenvolvimentos que envolvam um sistema supervisório de controle e aquisição de dados necessitam de IHMs para um processo de supervisão. Esse tipo de interface é muito semelhante a outro tipo de IHM, como a IHM *Embedded*, pois apresentam características que concordam até um determinado ponto. A diferença está no desenvolvimento da programação da IHM, além do valor necessário a ser imposto para a sua implementação [20].

As IHMs para supervisão têm aplicação baseada em computadores, normalmente desenvolvidas separadamente utilizando software com licença própria, enquanto o PC é apenas um hospedeiro que executa a aplicação, possibilitando muito mais recursos que sistemas embarcados.

A Sociedade Internacional da Automação propõe um ciclo de vida para as IHMs, de forma a estabelecer diretrizes que buscam padronizar as interfaces ao mesmo tempo em que

as tornam mais eficientes, intuitivas e de fácil compreensão. Isto posto, a padronização do design, da funcionalidade, do display e da interação entre os operadores e as IHMs fica a cargo desse órgão regulamentador.

2.2 Robótica Assistiva

As tecnologias assistivas foram arquitetadas a priori para executar tarefas comuns e rotineiras, permitindo que o ser humano dedicasse mais tempo a tarefas mais complexas. Pesquisadores e inovadores da área técnica estão constantemente desenvolvendo novas tecnologias e aprimorando-as, e os robôs por sua vez estão assumindo cada vez mais tarefas médicas e de cuidados.

As soluções mais comuns orientadas a serem aplicadas na navegação autônoma de robôs, localização visual simultânea e mapeamento de uma cena são ligeiramente focadas no entendimento ambiental baseado em características geométricas das imagens analisadas. A localização e o mapeamento semânticos simultâneos, caracterizados pela percepção ambiental em maior nível, promovem a aplicação da semântica do conteúdo para estimar posições com eficiência, como também construir mapas em várias dimensões, por exemplo. Dessa forma, uma análise semântica conseguirá promover insumos para a elaboração de tecnologias eficientes que estimulem a relação homem-máquina.

Pode-se dizer, portanto, que o uso de dispositivos robóticos se concentra em três áreas principais: Ajudar a monitorar o comportamento e a saúde das pessoas, ajudar pessoas que precisam de auxílio nas tarefas e situações diárias, e proporcionar interações sociais.

A especialista independente das Nações Unidas, no relatório da Organização das Nações Unidas de 2017, mostra que áreas relacionadas a tecnologia tocam inevitavelmente o proveito das pessoas idosas de seus direitos humanos, incluindo sua dignidade e autonomia, autodeterminação informacional e não discriminação e igualdade.

A menção acima compactua, ainda, com a necessidade de uma abordagem baseada nos direitos humanos para apoiar as discussões nesse domínio de assistência às pessoas em necessidade, além de garantir que sejam abordados adequadamente os desafios atuais e futuros e assegurem uma proteção suficiente dos direitos das pessoas que necessitam das tecnologias de forma mais ativa.

Para que haja uma maior capacidade de suporte a essas pessoas, o envolvimento da tecnologia faz-se necessário para impulsionar os resultados neste âmbito. Vários projetos

envolvendo Tecnologias Assistivas fazem uso de acessórios ou dispositivos extras, que irão fornecer dados um certo software ou serão usados para algum tipo de assistência ao usuário.

Um exemplo é o sistema desenvolvido em [21], onde diferentes sensores são utilizados como acessórios e auxilia pessoas com limitações visuais durante suas locomoções. Propostas utilizando técnicas de VC surgem como alternativa para eliminar a necessidade de dispositivos tecnológicos além do computador, aumentando a possibilidade de propagação dessas tecnologias.

Em [22], os autores do projeto projetaram um sistema capaz de controlar uma cadeira de rodas elétrica a partir de gestos com a cabeça. Na arquitetura do sistema, se utilizava um classificador em cascata para detectar a face e a técnica de Template Matching na região do nariz para reconhecer a posição do rosto. O usuário conseguia aumentar a velocidade, frear e virar movimentando a cabeça para a direção associada a cada ação.

2.3 A percepção na Robótica

A Percepção é uma das tarefas mais importantes para qualquer tipo de robô. Adquirir conhecimento sobre o ambiente envolve obter dados e medidas utilizando vários sensores, ou alguma forma de entrada do sistema, e posteriormente extrair informação útil desses dados coletados. [23]

Robôs que focam em atividades assistivas fazem uso de sensores e diversas técnicas de aprendizagem de máquina para realizarem as tarefas especializadas a que foram atribuídas com alta taxa de sucesso. Em muitas das soluções de robôs assistivos socialmente utilizando VC, as plataformas robóticas utilizadas fazem uso de câmeras para identificar humanos, objetos e textos. Alguns exemplos são: HelpMate, Transition Research Corp, USA, BibaBot, Bluebotics, Suíça.

O campo de Visão Computacional dá ao robô a capacidade de observar o ambiente de trabalho e extrair dados dele. A VC já é empregada em aplicações robóticas e não robótica, uma vez que engloba um grande número de diferentes técnicas no campo da engenharia óptica, processamento de vídeo, processamento digital de imagens, reconhecimento de padrões, inteligência artificial e computação gráfica

O processo básico da VC baseia-se no modelo da visão humana e interação olho-cérebro, também conhecido por *mammalian visual process.* A partir dessa inspiração

para a configuração desse campo da Inteligência Artificial, foi possível desenvolver grandes mudanças na forma de interpretação e extração de características de uma imagem.

2.4 Detecção de objetos em um contexto

Pode-se usar uma variedade de técnicas para realizar a detecção de objetos. Abordagens populares baseadas em aprendizado profundo usando redes neurais convolucionais (CNNs), como R-CNN e YOLO v2, aprendem automaticamente a detectar objetos em imagens.

Pode ainda ser escolhido duas abordagens principais para começar com a detecção de objetos usando o aprendizado profundo:

1. Criar e treinar um detector de objetos personalizado

Para treinar um detector de objetos personalizado do zero é preciso projetar uma arquitetura de rede para aprender os recursos dos objetos de interesse. Também é necessário compilar um conjunto muito grande de dados rotulados para treinar a CNN. Os resultados de um detector de objetos personalizado podem ser notáveis. Dito isso, é requerido configurar manualmente as camadas e pesos na CNN, o que requer muito tempo e dados de treinamento.

2. Usar um detector de objetos pré-treinado

Muitos fluxos de trabalho de detecção de objetos que usam aprendizado profundo aproveitam o aprendizado de transferência, uma abordagem que permite começar com uma rede pré-treinada e depois ajustá-la para a aplicação proposta. Esse método pode fornecer resultados mais rápidos porque os detectores de objetos já foram treinados com uma grande quantidade de dados.

2.5 A periculosidade em um cenário

Em [24], se utiliza o ambiente hospitalar para aplicar a utilização de VC na identificação de itens perigosos. Os vídeos usados para anotação foram gravados durante cirurgias realizadas no Departamento de Cirurgia da Universidade Médica Feminina de Tóquio de 2019 a 2020.

As imagens endoscópicas abdominais foram cortadas de vídeos cirúrgicos capturados manualmente para o modelo de treinamento, e imagens adicionais foram cortadas de outros vídeos para validação. As imagens variam em natureza, representando diferentes cirurgias e imagens duplicadas foram excluídas da avaliação. Finalmente, qualquer quadro de um vídeo no conjunto de treinamento foi excluído do conjunto de testes, considerando exclusiva em suas etapas a separação da base.

Uma estrutura de programação de código aberto para Redes Convolucionais foi usada para projetar um modelo que pudesse reconhecer e segmentar objetos em tempo real por meio do IBM Visual Insights. O modelo foi usado para detectar o trato gastrointestinal, sangue, vasos, útero, fórceps, portas, gazes hospitalares e clipes nas imagens cirúrgicas.

Como as abordagens habilitadas para IA podem processar grandes quantidades de dados cirúrgicos, elas podem ser usadas para reconhecer ou prever eventos adversos, permitir a "navegação" na cirurgia abordando várias questões de orientação anatômica e importantes ferramentas de tomada de decisão e contribuir para o treinamento e a educação.

Num contexto habitacional, o ambiente doméstico pode tornar-se um ambiente disseminador de doenças e agravos à saúde, sendo considerado um lugar de risco elevado para acidentes, especialmente nas idades entre 1 e 5 anos de idade, uma vez que, contém instrumentos que possuem algum aspecto chamativo para essa idade, como fósforos, garrafas de detergentes e materiais cortantes e brilhosos, além dos móveis e janelas e uma arquitetura nem sempre projetada para prevenir acidentes domésticos em crianças [25].

Estatísticas relatadas em estudos destacam que locais intradomiciliares, como cozinha e sala, apresentam uma maior taxa de acidentes [26]. A cozinha por ser um ambiente com vários utensílios e situações que podem levar a um incidente, como objetos perfurocortantes, panelas com o cabo para fora, fogo e outros. Já na sala, foi identificado instalações elétricas e eletrodomésticos ao alcance das crianças, além de cadeiras ou sofás que levam a queda das mesmas.

Todavia, é importante destacar que o acidente doméstico envolvendo especificamente crianças têm causas e consequências complexas, pois, além de envolver o ambiente, também possui como fatores determinantes os aspectos relativos ao indivíduo cuidador, se for o caso, como também à família e à própria criança [27].

3 Inteligência Artificial

A Inteligência Artificial é uma área da computação voltada a desenvolver algoritmos e sistemas capazes de realizar tarefas que demandam habilidades associadas à inteligência humana [28]. É possível entender esse nicho de estudo como um conceito que possibilita a criatividade dos seus autores no que tange o desenvolvimento de técnicas e soluções que enfrenta-se no dia-a-dia.

A capacidade de aprendizado pela máquina ou sistema computacional permite que o processo de desenvolvimento e aperfeiçoamento, fique por conta da observação e da qualidade do conjunto de dados que serão utilizados para treinar e melhorar a assertividade das decisões que serão ofertadas ao que se está desenvolvendo, sendo essas definições validadas após ao treino do algoritmo. [29] Sendo assim, a inteligência do sistema estará diretamente interligada à qualidade dos dados que estão sendo utilizados e dos exemplos cujo informações vão se reproduzir a partir do conhecimento que será extraído desses dados.

Com uma grande quantidade de conteúdo selecionado e passível de utilização para treinamento, as ferramentas de IA podem criar variações significativas de conteúdo personalizado. Um sistema que se baseia nos conceitos de IA pode avaliar as preferências comportamentais, cognitivas, ambientais, como até mesmo de engajamento e alinhar o aprendizado com os resultados de uma ação. A IA, dessa forma, facilita o aprendizado por meio de métodos de tentativa e erro [30], ou seja, quanto mais dados confiáveis disponíveis para treinar o algoritmo, melhor será o modelo gerado por ele.

3.1 Redes Neurais Artificiais

Na definição clássica trazida em seu livro, Haykin disserta que uma rede neural é um processador que tem a propensão natural de armazenar conhecimento experimental e torná-lo disponível para uso [31], ou seja, pode apresentar a habilidade e capacidade de aprender, assimilar, errar e, com esta experiência, adquirir novos conhecimentos, conferindo-lhe portanto, inteligência, semelhante ao cérebro humano. De acordo com o mesmo pesquidador, uma Rede neural artificial (RNA) se assemelha ao cérebro em dois pontos: No conhecimento adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem e na

forças de conexão entre neurônios - ou pesos sinápticos - são usadas para armazenar este conhecimento.

Esse processo de aprendizagem, reportado como primeira característica a se assemelhar com o órgão do corpo humano, quando dentro de contextos artificiais, consegue ser dividido em três tipos de aprendizado [32].

1. Aprendizado Supervisionado

Dentro do aprendizado supervisionado, se coleta um grande número de casos que possuem a entrada e a saída que se deseja. Essa entrada e saída são analisadas e produzem uma saída para novos casos. O uso de conjunto de dados rotulados utilizados são destinados a treinar e supervisionar algoritmos na classificação de dados e prever saídas da forma mais precisa possível. Usar entradas e saídas rotuladas ajuda o modelo a medir a acurácia do aprendizado e entender como melhorar esse aspecto.

Os problemas a serem resolvidos pelo conceito de Aprendizado Supervisionado pode ser dividido em dois: Problemas de classificação onde se usa um algoritmo para atribuir dados de teste com precisão em categorias específicas, como separar gatos de cachorros. Classificadores lineares, máquinas de vetor de suporte (SVM), árvores de decisão (DT) e floresta aleatória são tipos comuns de algoritmos de classificação. Já problemas de regressão utiliza o algoritmo para entender a relação entre variáveis dependentes e independentes, sendo utilizados largamente na previsão de valores de pontos de dados diferentes. Alguns algoritmos de regressão conhecidos são a Regressão Linear, Regressão Logística e Regressão Polinomial.

2. Aprendizado Não-supervisionado

Na aprendizagem não supervisionada, para encontrar os subgrupos de dados originais, se procura os diferentes padrões nos dados não rotulados.

O aprendizado não supervisionado usa algoritmos de Inteligência Artificial para analisar e agrupar conjuntos de dados não rotulados. A auto-organização demonstrada em redes neurais não supervisionadas, envolve o processo de competição e o processo de cooperação entre os neurônios da rede [33].

Esses algoritmos irão descobrir padrões ocultos nos dados sem a necessidade de intervenção de seu desenvolvedor e irão produzir resultados que associam os dados analisados ao contexto no qual ele foi testado.

3. Aprendizado por Reforço

Quando o algoritmo é inserido dentro da tentativa de aprendizado por reforço, ele não recebe uma resposta considerada correta, mas sim recebe um sinal de reforço, de recompensa [32]. O algoritmo fará dessa forma uma hipótese baseada nos exemplos e determina se essa hipótese foi boa ou ruim. Essa técnica é escolhida para ser utilizada em contextos de desenvolvimento de Jogos e Robótica.

Existe ainda dentro do Aprendizado por reforço o conceito de Dependência Temporal, que fará com que o algoritmo atrase o seu resultado ou feedback. Dessa forma, uma ação realizada em determinado momento não necessariamente terá uma recompensa imediata que determine o quão boa ela foi [34]. Isso atua na própria questão do aprendizado, pois o algoritmo irá se basear no histórico responsável por aquela recompensa, e não apenas em sua última ação. Consequentemente, sua tomada de decisão buscará o crescimento de sua assertividade, avaliando todas as ações realizadas.

A principal distinção entre as duas primeiras abordagens dissertadas acima é o tipo de conjunto de dados. Resumidamente, o aprendizado supervisionado usa dados de entrada e saída rotulados, enquanto um algoritmo de aprendizado não-supervisionado, não.

3.1.1 Perceptrons

O Perceptron nada mais é que a arquitetura mais simples de uma rede neural. Foi desenvolvido pelo cientista Frank Rosenblatt na Universidade de Cornell, nas décadas de 1950 a partir do artigo *The perceptron: A probabilistic model for information storage and organization in the brain* [35], publicado em um jornal de divulgação científica. Este artigo permite uma introdução clara dos conceitos matemáticos utilizados em redes neurais e como ela se comporta em termos mais práticos. um modelo probabilístico para armazenamento e organização de informações no cérebro

Seu funcionamento utiliza o seguinte fluxo: O modelo matemático irá receber diferentes entradas e produzir uma saída binária. Frank Rosenblatt introduziu os conceitos de pesos e de threshold, ou seja, variáveis adicionais para calcular essa saída. À vista disso, o resultado seria a soma ponderada desses pesos com os valores de entrada, sendo maiores ou menores que esse valor de threshold, ou valor limiar da rede.

Agindo como um classificador linear binário, um neurônio artificial pode representar essa arquitetura simples de uma rede neural, com uma camada única. De acordo com seus

fundamentos matemáticos, de forma mais detalhada, esse neurônio mapeará um vetor de entrada $x = [x_1, x_2, ..., x_m]$ a partir de uma função binária y, sujeitando este vetor x a uma soma ponderada com os pesos de sinapse $\theta = [\theta_1, \theta_2, ..., \theta_m]$ que flutuam no decorrer do aperfeiçoamento do modelo, acrescida de um valor de *bias b [36]* que influenciará no seu nível de tendenciosidade, proporcionando a função soma s abaixo, conforme equação 2.1 [35].

$$s = \sum_i x_i \theta_i + b_i \tag{2.1}$$

O fisiologista e pesquisador Warren McCulloch em seu artigo interpretou o funcionamento do neurônio biológico como sendo um circuito de entradas binárias combinadas por uma soma ponderada produzindo um valor efetivo [37].

Essa arquitetura, baseando-se no cérebro humano como inspiração, é constituída de:

- Entrada (remetendo-se aos dendritos) - onde são aplicados os sinais;
- Pesos (remetendo-se as sinapses) - onde fica retido o conhecimento;
- Função soma (remetendo-se a ação do processamento) - somatório da relação dos sinais de entrada e dos pesos sinápticos;
- Função de ativação (remetendo-se ao filtro do processamento) - função que dependendo do valor do somatório irá ou não ativar a saída, dependendo do valor de threshold escolhido.
- Saída (remetendo-se ao axônio) - saída de interface.

Ainda em [37], McCulloch e Pitts sintetizaram um modelo assumindo que os nós em cada camada da rede disparam simultaneamente, isto é, todos os nós ou neurônios artificiais são avaliados ao mesmo tempo e também as entradas em um instante de tempo x produzem a sua saída no tempo x+1. Sabe-se que não existe um mecanismo para sincronizar as ações dos nós receptores de conhecimento quando tratados na Biologia Humana, como também não há restrição para que as suas saídas sejam ativadas em tempos discretos como no modelo MCP. Sabe-se também que o valor da próxima saída dos neurônios biológicos depende enormemente das construções de ativações dos estados anteriores, já que até mesmo os neurotransmissores liberados anteriormente levam algum tempo para se recombinar, influenciando, assim, as ativações seguintes, como um efeito cadeia.

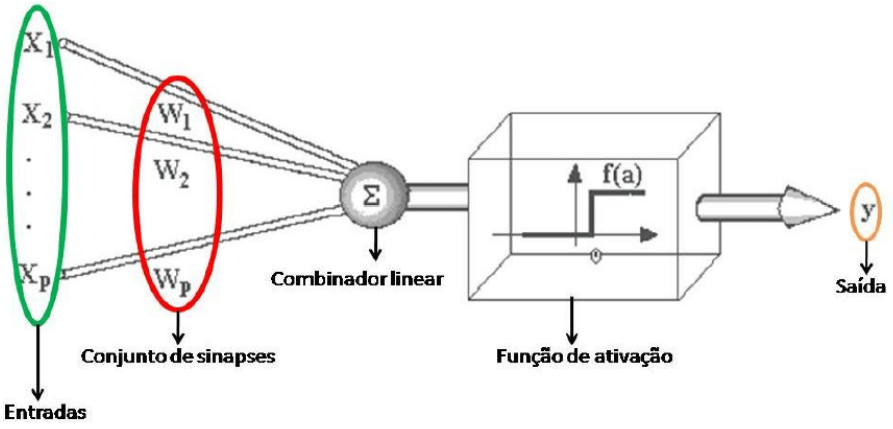
O resultado da operação somatória será então submetida a uma função de ativação, nesse caso do Perceptron é utilizada a função degrau, mapeando-se em uma saída binária conforme equação 2.2 abaixo. O perceptron atua, dessa forma, como um classificador linear, categorizando os dados de entrada em duas classes através de um hiperplano delimitado por s = 0. Caso os pesos θ associados aos respectivos valores de entrada sejam grandes o suficiente para alcançar o valor de ativação da função degrau, a sinapse é disparada e o resultado do modelo será y = 1.

$$y = \begin{cases} 0, & para \quad s < 0 \\ 1, & para \quad s \geq 0 \end{cases} \tag{2.2}$$

Quando dados não podem ser linearmente separáveis, é necessário o uso de modelos mais poderosos para realizar tais operações. Nesta situação é possível combinar *perceptrons* em uma estrutura em camadas, cada uma contendo uma quantidade diferente de neurônios.

Forma-se assim uma rede neural denominada *perceptrons* multicamadas (MLP), usada no contexto deste projeto como um dos componentes que constituem uma rede geradora adversária [38]. Em um perceptron de várias camadas, o vetor *x* é inserido na camada inicial ou *input layer,* cujos valores de saída servem como entradas da camada seguinte, construindo assim as camadas ocultas ou *hidden layers*. Essa ação se repete em camadas seguintes, seguindo uma estrutura totalmente conectada, na qual todos os neurônios de uma camada são ligados aos neurônios da seguinte até resultarem na camada de saída ou *output layer*. A escolha dos parâmetros do algoritmo de treinamento e funções de ativação é um passo de grande impacto na performance do sistema resultante [39].

Figura 1 - Classificação da Arquitetura



FONTE: CARVALHO, 2009.

O modelo apresentado na figura acima inclui em sua arquitetura um viés empregado, sendo representado pelo nó rotulado de combinador linear. Esse viés tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação explicada no último tópico, permitindo uma melhor adaptação do sistema ao conhecimento fornecido no decorrer do tempo.

A função de ativação, portanto, transforma as entradas recebidas em uma informação de saída e a envia para outro neurônio que a utiliza como informação de entrada, e dessa forma e a partir da sua replicação é criado as camadas do modelo neural.

Matematicamente um neurônio artificial e seu entorno é descrito a partir das seguintes equações abaixo:

$$u = \sum_{j=1}^{p} w_j x_j \tag{2.4}$$

$$v = f(u + b) \tag{2.5}$$

$$a = u + b \tag{2.6}$$

Sendo *x1, x2, ... , xj* os sinais de entrada; *wI, w2, ... , w*j os pesos sinápticos do neurônio; *u* a saída do combinador linear, ou função somatório, calculado devido aos primeiros sinais, e dessa forma e a partir da sua replicação é criado as camadas do modelo aos sinais de entrada; *b* sendo o viés; a o somatório entre a saída do combinador linear e do viés; *f(u+b)* é a função de ativação que recebe como entrada a; e *v* é o sinal de saída do neurônio.

A função de ativação irá converter a soma de ativação das entradas de saída. Esta função, que é um fator crucial de diferenciação entre os nós, define com base no potencial intrínseco de cada nó qual o valor que deve ser transmitido aos demais nós [40]. Ainda que a ação de cada nó em uma RNA seja independente da ação do outro, as ações são paralelas, e isso proporciona grande flexibilidade na construção do modelo.

Geralmente as funções de ativação são funções não-lineares, sendo as duas funções de ativação mais comuns a função threshold, usada em situações onde as entradas e saídas são binárias e a função sigmóide é a função mais comum utilizada em modelagem de redes neurais, por serem crescentes, contínuas e também diferenciáveis [39,40]. Determinando o nível de ativação de um neurônio, é possível limitar a amplitude do sinal de saída para uma faixa de valores finitos. Essa faixa geralmente é normalizada em um intervalo como [0,1] e [-1,1].

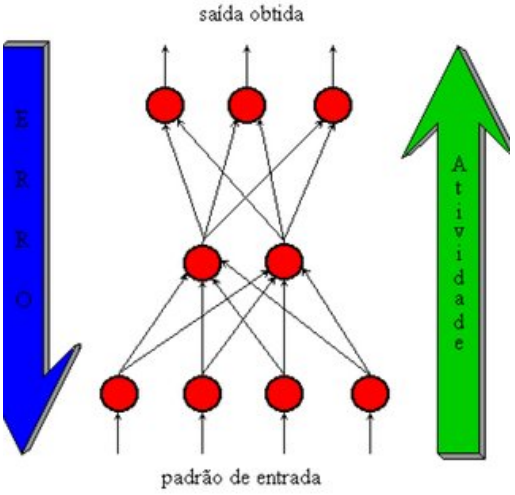
3.1.2 Backpropagation

Segundo Tonsig [41], o treinamento utilizando o algoritmo de *backpropagation* é dividido em 2 fases.

Na primeira parte dessa fase da aprendizagem, um padrão é apresentado à camada de entrada da rede. O resultado será propagado através da rede por cada camada, até que a resposta seja comparada à saída produzida pela última camada, sendo esta a camada de saída representada pelo *y* na Figura 1.

Na segunda parte, a mesma resposta de saída é comparada à saída desejada para esse padrão apresentado. Se a saída obtida não estiver correta, o erro será calculado a partir da propagação partindo da camada de saída até a camada de entrada e os pesos das conexões das unidades de camadas ocultas serão modificados conforme o erro passa pela retropropagação.

Figura 2 - Processo de aprendizado retroalimentado



FONTE: CARVALHO, 2009.

Depois que a rede estiver treinada, isto é, seu resultado seja o esperado, considerando o erro caso ele esteja em um nível considerado satisfatório, a mesma rede poderá ser utilizada como uma ferramenta para classificação de novos dados. Para que isso ocorra, a rede deverá ser utilizada apenas no modo progressivo, ou feed-forward. Dessa forma, novas entradas são apresentadas na camada de entrada, depois são processadas nas camadas intermediárias e os resultados são apresentados nas camadas de saída, como no treinamento, mas sem a retropropagação do erro relatada. A saída apresentada será o modelo dos dados, na interpretação da rede [42].

3.2 Redes Neurais Convolucionais

A Convolução é uma operação matemática entre duas funções f() e g(), construindo uma terceira função, que será o resultado da interação entre as duas funções. No processamento de imagens, ou PDI, a imagem é definida como uma função multidimensional [43], dessa forma, a convolução é útil para detecção e reprodução de algumas características, como bordas, suavização de imagem, extração de atributos, entre outras aplicações [44].

Sejam as funções f e g, ao receber-se uma variável contínua x, a convolução é definida como a seguinte estrutura:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) \bullet g(x - \tau)\, d\tau \tag{2.5}$$

O símbolo $*$ representa o operador de convolução, aplicada em diversos estudos e sistemas. Quando o x de entrada estiver definido no conjunto dos inteiros para a entrada das funções, a equação da convolução terá sua construção feita para se adaptar ao conteúdo discreto, isto é:

$$f[x] * g[x] = \sum_{n=-\infty}^{\infty} f[n] \cdot g[x-n] \tag{2.6}$$

Caso o sistema comporte duas variáveis x e y, as equações ficarão nos seguintes formatos:

$$f(x,y) * g(x,y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1,\tau_2) \cdot g(x-\tau_1, y-\tau_2) d\tau_1 d\tau_2 \tag{2.7}$$

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2] \tag{2.8}$$

A convolução de uma imagem é bem exemplificada quando há um somatório da multiplicação de cada elemento da imagem, junto com seus vizinhos locais, pelos elementos das listas de vetores que representam o filtro de convolução.

3.2.1 Estrutura das redes convolucionais

Uma rede convolucional é um perceptron multicamada projetado especificamente para reconhecer determinados padrões [38], sendo essa necessidade aprendida de forma supervisionada por meio de uma rede cuja estrutura inclui as seguintes formas de restrições

- **Extração de características:** cada neurônio que compõe a arquitetura irá receber sinais de entrada de um campo da camada anterior, impondo dessa forma a extração de características locais. Esta extração de características locais faz com que a posição exata de cada característica seja irrelevante, desde que sua posição em relação às características vizinhas permaneça sem alterações.

- **Mapeamento de características:** cada camada computacional da rede é composta por diversos mapas de características, que são regiões onde os neurônios têm como dever compartilhar os mesmos pesos sinápticos. Esses pesos são chamados de filtros ou *kernels*, e dão robustez ao modelo, fazendo com que ele seja capaz de lidar com variações de distorção, rotação e translação na imagem

- **Subamostragem:** após cada camada de convolução é aplicada uma camada de subamostragem (*subsampling*), que nada mais é que uma coleta de amostras de cada mapa de característica. Estas amostragens podem ser realizadas obtendo-se a soma, tirando a média, selecionando-se o maior (*max pooling*) ou menor (*min pooling*) valor da região em análise.

3.2.2 Camadas de Convolução

As camadas de convolução, em conjunção com outros tipo de camada, irão determinar uma arquitetura específica para uma finalidade. Uma rede neural profunda tem diferentes camadas dos neurônios artificiais, mas, de forma simplificada, sempre haverá uma camada de entrada, uma camada intermediária e uma camada de saída.

Em uma rede completamente conectada, como por exemplo em uma imagem 300x300, cada pixel seria diretamente ligado a cada neurônio na camada de entrada. Dessa forma, seria necessário uma exacerbada quantidade de parâmetro como 90 mil neurônios, desperdiçando recurso computacional.

Por isso, a aplicação de camadas convolucionais sobre os pixels da imagem, reduz muito a quantidade de parâmetros e facilita a descoberta de padrões. Geralmente utiliza-se

filtros espaciais lineares como visto no artigo anterior. Uma camada convolucional realiza o aprendizado de múltiplos filtros, onde cada filtro, ou kernel, extrai uma informação da imagem.

Por exemplo, a seguir estão as camadas utilizadas para desenvolvimento de uma CNN:
Camada de Convolução: Nesta camada é realizada a convolução e deve-se especificar a quantidade de filtros, e o tamanho do núcleo que irá percorrer a imagem.

Camada de Ativação: É a camada responsável por se utilizar da função de ativação, ao final de uma camada de convolução e da camada densa. Esta função tentava representar o cenário da biologia onde acredita-se que os neurônios só conseguem propagar uma informação pelo axônio após alcançarem um limiar mínimo de estímulo, e nessa situação estariam ativados. Na prática, essa função de ativação busca inserir um comportamento de não-linearidade, após a função linear dos pesos com as entradas, do corpo da célula.

Camada de Pooling ou Agrupamento: É a camada que reduzirá as dimensões da imagem, buscando manter o número de canais ou a sua profundidade, sendo a entrada analisada de acordo com seus valores atuais. Essa camada busca tornar a rede convolucional invariante a uma transformação geométrica, ou seja, tornar a rede um classificador capaz de dizer se há a presença de um item na imagem, independentemente de onde ele esteja posicionado.

Camada Dropout: Camada utilizada com o propósito de regularização, onde alguns neurônios são desligados aleatoriamente, juntamente com suas conexões, durante o treinamento apenas. Durante a predição todos os neurônios são mantidos ativos. O motivo de se fazer isso é evitar sobreajuste ou *overfitting* no treinamento.

Camada Flatten: a matriz resultante das camadas anteriores de convoluções e poolings passa novamente pelo processo de dimensionamento para se tornar um array linear, de uma única dimensão, ou seja, um vetor de características. Esta etapa é um preparo para se entrar na camada principal da rede neural totalmente conectada.

Camada Dense: é implementada a rede neural totalmente conectada, ou *fully connected*, e deve-se informar a dimensão da saída e a função de ativação a ser utilizada, sendo elas conectadas por pesos, sendo a saída um vetor com a distribuição de probabilidades para cada classe.

Algumas principais funções de ativação que são utilizadas para CNNs são as seguintes: Sigmóide (utilizada para aprendizado de funções lógicas, pois limita os valores entre 0 e 1), Tangente Hiperbólica (utilizada em redes de memória de curto prazo longa, uma rede neural recorrente, esta função limita os valores entre -1 e 1), ReLU ou Unidade Linear Retificada (utilizada para imagens, não satura na região positiva, converge mais rápido do que a sigmóide ou tangente hiperbólica sobre imagens, não sendo centrada em zero) e Softmax (produz uma distribuição das probabilidades para cada classe de imagem durante uma classificação, diferente da sigmóide capaz de lidar com apenas duas classes. É utilizada na camada de saída da CNN).

A seleção de hiper-parâmetros é feita de acordo com a camada às quais pertencem. Para as camadas de convolução, por exemplo, os hiperparâmetros selecionados podem ser a quantidade e tamanho dos filtros,o stride e zero-padding. Estes parâmetros influenciarão propriamente na extração das características, O tamanho do filtro e de zero-padding impactarão diretamente na área de busca. Já o stride indica o tamanho do passo da janela do filtro sobre a entrada da camada atual.

Os hiperparâmetros escolhidos variam de acordo com a arquitetura e a base escolhidas. Para cada par de arquitetura e base, um conjunto de hiper-parâmetros são escolhidos para as camadas de convolução e pooling. Estes hiperparâmetros podem se basear em outros trabalhos que utilizam redes neurais convolucionais como [45-47].

Ainda dependendo da escolha de hiperparâmetros, uma camada pode resultar em uma saída inválida. Para isso, basta que os hiperparâmetros escolhidos em uma camada resultem em um volume nulo ou negativo, neste caso, considera-se que a operação impossível de ser realizada. [48]

3.2.4 Arquitetura

A escolha da arquitetura da rede é feita de acordo com o seguinte critério: erros de classificação, performance e a complexidade de aprendizagem da rede neural [49].

Segundo André Ponce [50], professor e pesquisador da Universidade São Paulo, as etapas de desenvolvimento de aplicações de redes neurais podem ser divididas da seguinte forma: Coleta de dados e separação em conjuntos, Separação dos dados, Configuração da rede, Treinamento, Teste, Integração. A seguir será listado uma breve descrição de cada parte do desenvolvimento.

1ª parte: Coleta de dados e separação em conjuntos

Os dois primeiros passos do processo de desenvolvimento de redes neurais artificiais são a coleta de dados relativos ao problema e a sua separação em um conjunto de treinamento e um conjunto de testes. Além disso, os dados coletados devem ser significativos e cobrir amplamente o domínio do problema.

2ª parte: Separação dos Dados

Normalmente, os dados coletados são separados em duas categorias: dados de treinamento, que serão utilizados para o treinamento da rede e dados de teste, que serão utilizados para verificar sua performance sob condições reais de utilização. Além dessa divisão, pode-se usar também uma subdivisão do conjunto de treinamento, criando um conjunto de validação, utilizado para verificar a eficiência da rede quanto a sua capacidade de generalização durante o treinamento, e podendo ser empregado como critério de parada do treinamento. Esses conjuntos são geralmente colocados em ordem aleatória para prevenção de tendências associadas à ordem de apresentação dos dados. Além disso, pode ser necessário pré-processar estes dados, através de normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização na rede.

3ª parte: Configuração da rede

Este passo pode ser dividido em três etapas: Seleção do paradigma neural apropriado à aplicação, determinação da topologia da rede a ser utilizada (como o número de camadas, o número de unidades em cada camada) e a determinação de parâmetros do algoritmo de treinamento e funções de ativação. É comum as escolhas desses fatores serem feitas de forma empírica, ainda que existam metodologias na condução destas tarefas.

4ª parte: Treinamento

Nesta fase, seguindo o algoritmo de treinamento escolhido, serão ajustados os pesos das conexões. É importante considerar, nesta fase, alguns aspectos tais como a inicialização da rede, o modo de treinamento e o tempo de treinamento.

Uma boa escolha dos valores iniciais dos pesos da rede pode diminuir o tempo necessário para o treinamento. Normalmente, os valores iniciais dos pesos da rede são números aleatórios uniformemente distribuídos, em um intervalo definido. Quanto ao modo de treinamento, na prática é mais utilizado o modo padrão devido ao menor armazenamento de dados, além de ser menos suscetível ao problema de mínimos locais, devido à pesquisa de

natureza estocástica que realiza. Por outro lado, no modo batch se tem uma melhor estimativa do vetor gradiente, o que torna o treinamento mais estável. A eficiência relativa dos dois modos de treinamento depende do problema que está sendo tratado.

Vários fatores podem influenciar a duração do tempo de treinamento, porém sempre será necessário utilizar algum critério de parada. O critério de parada do algoritmo *backpropagation* não é bem definido, e geralmente é utilizado um número máximo de ciclos. Pode ocorrer que em um determinado instante do treinamento a generalização comece a degenerar, causando o problema de over-training, ou seja a rede se especializa no conjunto de dados do treinamento e perde a capacidade de generalização.

5ª parte: Teste

Durante esta fase o conjunto de teste é utilizado para determinar a performance da rede com dados que não foram previamente utilizados. A performance da rede, medida nesta fase, é uma boa indicação de sua performance real.

Devem ser considerados ainda outros testes como análise do comportamento da rede utilizando entradas especiais e análise dos pesos atuais da rede, pois se existirem valores muito pequenos, as conexões associadas podem ser consideradas insignificantes e assim serem eliminadas (*pruning*).

De modo inverso, valores consideravelmente maiores que os outros poderiam indicar que houve treinamento exacerbado (*overtraining)* da rede.

6ª parte: Integração

Finalmente, com a rede treinada e avaliada, ela pode ser integrada em um sistema do ambiente operacional da aplicação. Para maior eficiência da solução, este sistema deverá conter facilidades de utilização como interface conveniente e facilidades de aquisição de dados através de planilhas eletrônicas, interfaces com unidades de processamento de sinais, ou arquivos padronizados. Uma boa documentação do sistema e o treinamento de usuários são necessários para o sucesso do mesmo.

Além disso, o sistema deve periodicamente monitorar sua performance e fazer a manutenção da rede quando for necessário ou indicar aos projetistas a necessidade de retreinamento. Outras melhorias poderão ainda ser sugeridas quando os usuários forem se tornando mais familiares com o sistema, estas sugestões poderão ser muito úteis em novas versões ou em novos produtos.

4 Processamento de Conteúdo

4.1 Conjuntos de dados

A aplicabilidade das abordagens de deep learning pode ser vista pela utilidade em vários conjuntos de dados, sendo esses dados moldados pelo desenvolvimento da sociedade, de acordo com o cenário de aplicação. Imagens são conteúdos bastante utilizados para ser um domínio de uma análise de caso, ou seja, a peça-chave para o desenvolvimento de novos conteúdos em ambientes corporativos ou acadêmicos.

Uma breve descrição de alguns conjuntos de dados existentes para fins de análise de áreas de utilização é fornecido abaixo.

1. Imagens em tons de cinza: O conjunto de dados de imagens em tons de cinza mais usado é MNIST [45] e suas variações, como NIST. O cenário de aplicação é o reconhecimento de dígitos escritos.

2. Imagens de Características Faciais: O conjunto de dados Adience [51] pode ser usado para identificação de atributos faciais, que é, idade e gênero, a partir de imagens de rostos. Ele é composto por 26.580 fotos que possuem um rótulo de gênero binário e um rótulo de oito faixas etárias diferentes, divididos em cinco divisões.

3. Imagens Médicas: O conjunto de dados de *Chest X-ray* [52] compreende 112120 imagens de raios-X de visão frontal de mais de 30 mil diferentes pacientes com os rótulos de imagem de quatorze doenças diferentes, onde cada imagem pode conter um ou mais rótulos. O conjunto de dados Lymph Node Detection and Segmentation [53] também podem ser citados nesse âmbito por serem formados por imagens de tomografias computadorizadas do mediastino e do abdômen.

4. Fluxos de vídeo. O conjunto de dados WR [54, 55] pode ser usado para reconhecimento de atividade baseado em vídeo em linhas de montagem [56], contendo sequências de 7 categorias de atividades industriais. Já o YouTube-8M [57] é um conjunto de dados de 8 milhões de vídeos do YouTube URLs, juntamente com rótulos de nível de vídeo de um conjunto diversificado de 4800 diferentes entidades que se baseiam em gráficos de conhecimento, ou *Knowledge Graph*.

As CNNs têm a capacidade única para aprendizagem de recursos, ou seja, de recursos de aprendizado automático com base no conjunto de dados fornecidos. CNNs também são

invariantes para formações, mas por outro lado, são fortemente dependentes da existência de dados rotulados.

4.2 Deep Learning

Ainda que redes neurais conseguissem ter um impacto bastante significativo em muitas aplicações, o interesse em pesquisar esse tema começou a ser concentrado em um outro assunto baseado no conceito de RNA.

Com a introdução do conceito de *Deep Learning* (DL) por Hinton et al [58] e o tópico ficando cada vez mais proeminente, resultou-se então em um renascimento na pesquisa rede, sendo esta época referida em alguns momentos como "nova geração de redes neurais". Isso ocorre porque as redes profundas, quando devidamente construídas e treinadas, têm produzido sucesso significativo em uma variedade de desafios de classificação e regressão [59].

Atualmente, a tecnologia DL é considerada uma das mais recomendadas em ser utilizada e estudada na área de aprendizado de máquina, inteligência artificial, ciência e análise de dados, uma vez que se leva em consideração suas capacidades de aprendizagem dos dados fornecidos. Muitas corporações incluindo Google, Apple, Microsoft, Meta etc., estuda esse tema ativamente pois conclui que pode fornecer resultados consideráveis em diferentes problemas de regressão e classificação, tal qual na análise e extração de informação de conjuntos de dados [59].

Em termos de domínio de trabalho, DL é considerado como uma subárea ou subconjunto do nicho de Machine Learning e Inteligência Artificial e, portanto, DL pode ser visto como uma função de AI que se inspira no processamento de dados do cérebro humano.

A raiz do DL está aumentando dia após dia, o que é mostrado em certos artigos como [60] com base nos dados históricos coletados do Google Trends [61]. O aprendizado profundo é diferente do aprendizado de máquina padrão em termos de eficiência, uma vez que considera a oscilação com o volume de dados gerados e requeridos.

A tecnologia de DL usa várias camadas para representar as abstrações de dados para construir modelos computacionais. Ainda que modelos baseados nesta área levam um tempo considerável para treinar um modelo devido a um grande número de parâmetros, leva-se um curto período de tempo para serem executados na fase de testes em comparação com outros algoritmos de aprendizado de máquina [62].

Uma rede neural típica é composta simplesmente por pontos de comunicação básicos e conectados, formado por elementos que irão processar a informação chamados neurônios,

cada um dos quais gera uma série de ativações de valor real para o alvo resultante.

A Figura 1 ainda mostra uma representação esquemática do modelo matemático de um neurônio artificial, ou seja, elemento principal do , destacando a entrada (Xi), peso (w), viés (b), soma função de comunicação (∑), função de ativação (f) e sinal de saída correspondente (y). Hoje, a tecnologia DL baseada em rede neural é amplamente aplicada em muitos campos e áreas de pesquisa, como saúde, análise de sentimentos, PLN, reconhecimento visual, inteligência de negócios, segurança cibernética, entre outros.

Embora os modelos de DL sejam aplicados com sucesso em várias áreas de aplicação, como as mencionadas acima, construir um modelo adequado de deep learning é uma tarefa que desafia quem a está desenvolvendo, uma vez que deve ser considerado a natureza dinâmica e as variações dos problemas do mundo real e dados.

Além disso, os modelos DL são normalmente considerados como dispositivos sigilosos que dificultam o desenvolvimento e padronização de pesquisa e aplicação de aprendizado profundo As técnicas de DL podem desempenhar um papel fundamental na construção de sistemas inteligentes orientados a dados de acordo com as necessidades atuais, por causa de seus excelentes recursos de aprendizado a partir de dados históricos.

4.3 Detecção de Objetos

A detecção de objetos é o ato de detecção de instâncias de conteúdos semânticos de uma determinada classificação em imagens e vídeos digitais. Uma abordagem comum para estruturas de detecção de objetos inclui a criação de um grande conjunto de janelas candidatas que são classificadas na sequência usando recursos de uma Rede Convolucional.

Por exemplo, o método descrito por Girshick et al., emprega busca seletiva [63] para derivar propostas de objetos, extrai recursos CNN para cada proposta e, posteriormente, alimenta os recursos para um classificador SVM para decidir se as janelas incluem o objeto ou não. Um grande número de trabalhos é baseado no conceito de regiões com características CNN como discutido em [17].

Abordagens seguindo o paradigma de regiões com CNN geralmente têm boas precisões de detecção [64,65] no entanto, há um número considerável de métodos que tentam melhorar ainda mais o desempenho de regiões com abordagens CNN, onde alguns conseguem encontrar posições aproximadas dos objetos em análise, mas muitas vezes não podem determinar com precisão a posição exata dele [66].

Para este fim, tais métodos geralmente seguem uma abordagem de segmentação semântica de detecção conjunta de objetos [67-69], geralmente alcançando bons resultados. A grande maioria dos trabalhos sobre detecção de objetos usando deep learning aplica uma variação de CNNs, por exemplo, em estudos como [70-72] uma nova camada *def-pooling* (que busca aprender contextos de deformação de partes de objetos com diferentes tamanhos e significados semânticos) e uma nova estratégia de aprendizado são propostas, em [73] são utilizadas CNNs fracamente supervisionadas em cascata e em [74] CNNs com reconhecimento de subcategorias.

No entanto, existe um número relativamente pequeno de tentativas de detecção de objetos usando outros modelos profundos. Por exemplo, [75] propõe um método grosseiro de localização de objetos baseado em um mecanismo de características únicas (saliência) com um DBN para detecção de objetos em imagens de sensoriamento remoto; o estudo [76] apresenta um novo DBN para reconhecimento de objetos 3D, no qual o modelo de nível superior é uma máquina Boltzmann de terceira ordem, treinada usando um algoritmo híbrido que combina gradientes generativos e discriminativos; [77] emprega uma abordagem de fusões entre métodos de Deep Learning, enquanto [78] explora as capacidades de representação de um modelo profundo em um paradigma semi-supervisionado. Destarte, [79] aproveita a técnica de autoencoders (aprendizado não-supervisionado que aprende representação de dados eficientes a partir do treinamento que ignora ruídos) em sequência para detecção de múltiplos órgãos em imagens médicas, enquanto [80] explora autoencoders de forma empilhada guiadas por saliências para detecção de objetos baseados em vídeo.

5 Metodologia

O conjunto de atividades desenvolvidas foram condensados em quatro fases mais generalistas:

1. Planejamento: Fase composta pelo estudo do problema, definição do escopo, arquitetura, e ferramentas técnicas;
2. Absorção dos dados: Efetuar a organização e o recebimento dos conteúdos a serem analisados;
3. Desenvolvimento e execução: Implementação e testes das ferramentas;
4. Produção escrita: Fase composta pela escrita do texto final.

Para isso serão utilizadas a linguagem de programação *Python* [81], a ferramenta de edição de código e geração de resultados *Google Colab* e eventualmente as bibliotecas e ferramentas que entrarão em concordância com o desenvolvimento do código.

Por ser uma linguagem de programação amplamente utilizada no mundo, permitir o uso de interfaces, robustez e identificação de bugs em tempo de compilação, *Python* se torna uma opção factível pelo seu alto desempenho e escalabilidade. Uma grande vantagem dessa linguagem é do código escrito em Python ser de fácil utilização em qualquer ambiente de edição ou sistema operacional e também ser uma das principais escolhas de linguagem quando no campo de Visão Computacional.

Para reduzir a complexidade no código e facilitar a manutenção, a arquitetura será baseada em algumas bibliotecas que já existem no campo acadêmico e profissional como o *OpenCV* [82]. Por ser um dos padrões mais utilizados atualmente e por possibilitar a utilização de algumas funções e conhecimentos que já foram desenvolvidos e permitidos para uso, o uso desse projeto alavancará o tempo de desenvolvimento e também o tempo para testagem do resultado após concluída toda a análise.

A ferramenta *DarkLabel* [83] fará o processo de interpolação automática dos objetos em cena. Juntamente com a ferramenta de versionamento de código git, e plataforma de armazenamento de conteúdo Github, a gestão dos resultados recebidos pelo *DarkLabel* será feita de forma sincronizada com o repositório virtual da pesquisa.

Durante o desenvolvimento do projeto, serão abordados os conceitos de aprendizagem de máquina, com foco em deep learning; detecção e classificação de objetos em cenas naturais; análise semântica e processamento de linguagem natural e sua utilização em visão

computacional e robótica assistiva. Tende-se ainda a realização de modificações sobre modelos de estado-da-arte de Visão Computacional, uma vez que são capazes de operar sobre plataformas robóticas que comumente apresentam limitações consideráveis na construção de seu hardware.

Para a definição do conteúdo a ser classificado, foram utilizados alguns materiais. Sendo eles: tomadas (independente do formato e construção), facas (independente do tamanho, ou de onde ela é utilizada) e tesouras (independente de ser profissional, com ponta, tamanho ou cor).

Por fim, estimou-se que a geração de dados por cada vídeo de um usuário fique em torno de 40 minutos, somando todas as necessidades da pesquisa, incluindo as variações de horário para gravações tal qual construção do formato ideal.

5.1 Base de dados

Sendo uma nova base de dados a proposta do projeto aqui dissertado, a base de imagens utilizadas para teste e treinamento no trabalho foi construída a partir de vídeos gravados pelo autor. Para simular casos reais com a classificação de imagens de alta qualidade foram utilizados 35 vídeos que geraram cada um cerca de 3500 imagens contendo a classificação binária dos itens dispostos a cada *frame* do vídeo. Estes vídeos estão hospedados no Google Drive e passaram por um critério de conformidade com algumas métricas estipuladas para sua formação, explicadas a seguir no tópico 5.2.2.

5.2 Coleta de dados

Foi elaborado um formulário eletrônico disponibilizado por meio de um *link* do Google Forms para que os vídeos pudessem ser adicionados e posteriormente passados por uma classificação manual de validação. A página inicial do formulário de arrecadação dos dados está disponível no Apêndice.

Para garantir a seguridade do conteúdo compartilhado, também foi necessário os voluntários participantes concordarem com um termo de conformidade no ato de adição dos dados no formulário eletrônico, também disponível na seção de Apêndice.

5.2.1 Formato

Foram necessários vídeos em um formato específico, para padronizar a classificação. São eles:

- Os vídeos deverão ser gravados por uma câmera de celular
- A duração de cada vídeo será de aproximadamente 60 segundos
- A resolução tem de ser a mais próxima de 1280x960
- Os vídeos deverão ser gravados num intervalo de altura entre 55cm a 60cm.
- O vídeo deverá percorrer todo o ambiente escolhido linearmente

Essas restrições são necessárias uma vez que o produto final irá simular o ambiente visual de um Robô NAO [84], dispositivo abordado no projeto e que possui métricas necessárias para utilização de seu ambiente onde o trabalho está inserido.

5.2.2 Forma dos vídeos

Uma vez que buscava-se produzir 8 vídeos em cada período do dia percorrendo uma distância linear do ambiente escolhido pelo voluntário, a composição estrutural de cada bloco de criação foi sugerida da seguinte forma. No período de 1 dia, o usuário deveria gravar 24 vídeos de 1 minuto cada, movendo-se pelo seu ambiente expondo o conteúdo das cenas de forma clara para que a anotação fosse feita.

Tabela 1 - Tipos de vídeos necessários para formação da base

Quantidade	Item	Posição e situação de gravação
1	Faca	Em cima de um objeto de maior altura, percorrendo a distância padrão
1	Faca	Em cima de um objeto de média altura, percorrendo a distância padrão
1	Faca	Em cima de um objeto de baixa altura, percorrendo a distância padrão

1	Tesoura	Em cima de um objeto de maior altura, percorrendo a distância padrão
1	Tesoura	Em cima de um objeto de média altura, percorrendo a distância padrão
1	Tesoura	Em cima de um objeto de baixa altura, percorrendo a distância padrão
1	Tomada	Bloqueadas por protetores de tomada, ou um papel , percorrendo a distância padrão
1	Tomada	Disponíveis para uso, percorrendo a distância padrão

Fonte: O Autor

5.2.2.1 Ambientes para dispor os objetos

A escolha do ambiente foi aberta para dispor os objetos periculosos em qualquer posição de um dos conteúdos listados abaixo.

Tabela 2 - Objetos classificados por altura para formação da base

Altura do objeto	**Tipo de objeto**
Maior altura	Escrivaninha, armário aberto, prateleira
Média altura	Cama, cadeira
Baixa altura	Chão, tapete, suporte de calçados

Fonte: O Autor

Considerando que há variação da luminosidade do ambiente, foi indicado que os vídeos fossem feitos nos seguintes períodos:

Tabela 3 - Definição de horários para gravação dos vídeos

Período do dia	**Horário**
Manhã	8 horas ≤ período ≤ 10 horas
Tarde	15 horas ≤ período ≤ 17 horas
Noite	20 horas ≤ período ≤ 22 horas

Fonte: O Autor

Sendo os dois períodos iniciais do dia considerando a luz natural do ambiente, já o último, a luz artificial.

Na formação dos vídeos, ainda, se encorajou a presença de objetos não perigosos em posições diversas no vídeo, influenciando a classificação de tal objeto prioritariamente periculoso dentro de um cenário específico como seguro, uma vez que a situação de risco estaria reduzida dependendo da forma analisada.

5.2.3 Armazenamento e envio

Cada vídeo será considerado um arquivo a ser classificado. Ao final da coleta, seria encontrado 24 vídeos por voluntário, pois, em cada período do dia foi almejado a gravação de 8 vídeos.

A nomenclatura de arquivos de vídeo buscou seguir a sintaxe abaixo:

{criador}_{altura}_{periodo}_{hora}

Os vídeos deveriam ser salvos no formato mp4, zipados e enviados como anexo no formulário.

5.3 Anotação dos dados

5.3.1 Ferramenta de rotulagem e anotação de vídeo/imagem

DarkLabel [83] foi o programa utilizado para criação da base, sendo ele uma ferramenta de anotação de conteúdo baseado na linguagem Python. Seu objetivo é rotular caixas delimitadoras de objetos com ID e nome em vídeos e imagens, permitindo assim a construção da base. Ele também pode ser usado para cortar vídeos em diferentes períodos de tempo, mostrar imagens de treinamento em um vídeo e permitir que a região de imagem seja deferida como um mosaico deformável, garantindo a segurança do conteúdo.

5.3.1.1 Principais características

- O Darklabel é uma ferramenta que possui a capacidade de rotulagem automática de objetos por rastreamento visual (múltiplos alvos) e também rotulagem semiautomática por interpolação linear
- Também é possível dentro da ferramenta estipular teclas de atalho configuráveis pelo usuário e suporte para aumentar e diminuir o zoom
- Os formatos de dados são configuráveis pelo usuário (pascal voc, darket yolo, xml/txt, quaisquer outros formatos definidos pelo usuário)
- Ferramenta de divisão de vídeo (em imagens) e mesclagem de imagens (em um arquivo de vídeo), dispostas na interface da ferramenta
- Ferramenta de corte de vídeo (corte e salve apenas a seção selecionada no vídeo)
- Como falado anteriormente, também possui a ferramenta de mascaramento de privacidade de vídeo/imagem (mosaicar a área da caixa na imagem)
- É capaz de aplicar rastreamento de somente trajetórias selecionadas ou caixas recém-criadas

5.3.1.2 Características adicionais do Darklabel

- Amostragem/coleta de dados

Se o objetivo é separar imagens em um vídeo e salvá-las (por exemplo, coletando amostras de treinamento), faz-se necessário que se desenha as caixas fictícias nas imagens e exporte os resultados da anotação como imagens com a opção "sem desenho de caixa" selecionada e a opção "somente quadros rotulados" marcada .

- Mascaramento de privacidade

A ferramenta permite o preenchimento modificado definido como caixas na área de privacidade e exporte os resultados da anotação com a opção "mosaic the box área" selecionada. As caixas de marcação selecionadas seriam utilizadas para borrar rostos humanos garantindo a confidencialidade do processamento.

5.3.2 Configurações necessárias para a anotação

O programa pode ser configurado modificando *o arquivo darklabel.yml* anexado na aplicação após download. E as possibilidades de modificação no arquivo variam:

- É possível definir e modificar formatos de dados
- É possível definir e alterar teclas de atalho (navegação de quadros, teclas de ação)
- É possível definir e alterar a configuração de exportação de vídeo/imagem (codec de vídeo, taxa de quadros, formato de imagem)
- É possível definir o diretório padrão de salvar/carregar
- É possível definir rótulos de classe
- É possível definir ajustar o desenho da Interface do usuário.

Para a anotação dos dados da base, foi escolhido o formato XML para expressar os valores extraídos da imagem.

Para expressar a quantidade de classes e os tipos de classes que seriam utilizadas nas anotações, foi criado um array de conteúdos informado se o objeto detectado é seguro (safe) ou inseguro (unsafe).

No arquivo *darklabel.yml* então, as seguinte linhas foram definidas:

Tabela 4 - Configuração no arquivo .yml da ferramenta

classes_tcc_binary: ["safe", "unsafe"]
gt_file_ext: "xml"

Fonte: O Autor

Dessa forma, foi possível gerar arquivos XML que servirá para as anotações posicionais e informativas dos valores disponíveis para compartilhamento na base de rotulagem. Esses dados foram acessados utilizando *xml.etree.ElementTree* , um conversor *built-in* da própria linguagem para identificar e percorrer por valores dentro de um arquivo XML.

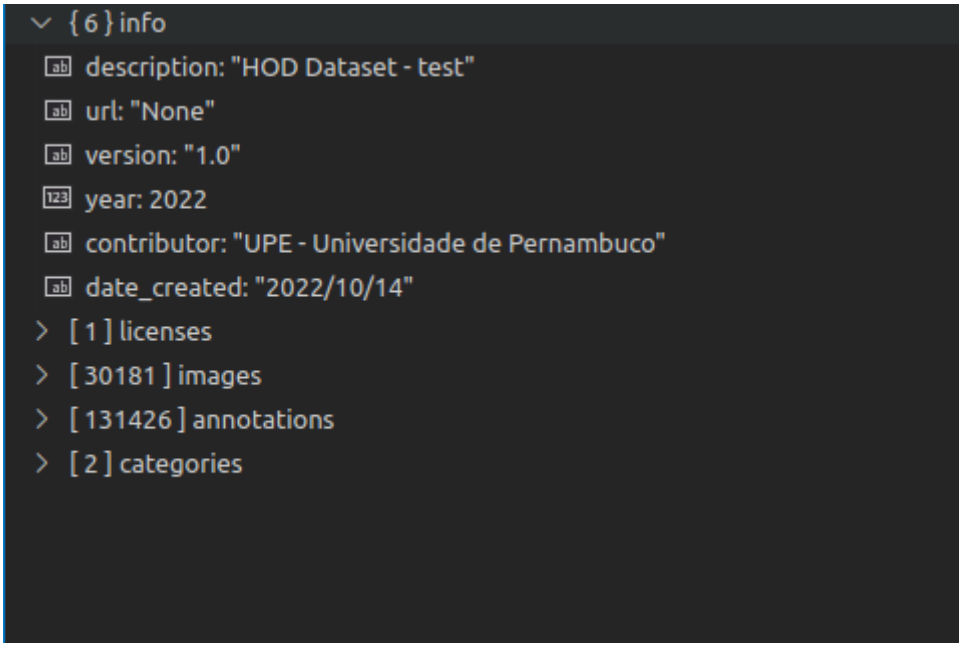
Figura 3 - Anotação gerada no formato XML

```
<annotation>
  <folder>Tarde</folder>
  <filename>Victor_60_tarde_1630.mp4</filename>
  <path>C:\users\richard\My Documents\TCC\Victor\Tarde\Victor_60_tarde_1630.mp4</path>
  <source>
    <database>undefined</database>
  </source>
  <size>
    <width>1280</width>
    <height>720</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
  ...
  </object>
  <object>
  ...
  </object>
  <object>
  ...
  </object>
  <object>
  ...
  </object>
  <object>
  ...
  </object>
  <object>
    <name>unsafe</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>413</xmin>
      <ymin>387</ymin>
      <xmax>473</xmax>
      <ymax>439</ymax>
    </bndbox>
  </object>
  <object>
    <name>safe</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>312</xmin>
      <ymin>333</ymin>
      <xmax>506</xmax>
      <ymax>407</ymax>
    </bndbox>
  </object>
</annotation>
```

Fonte: O Autor

A partir de uma interação com o código, foi feita a conversão das anotações XML geradas. Ou seja, a base também foi disponibilizada em um outro padrão: O padrão COCO (Common Objects in Context) [85], o mais comum e utilizado em problemas de detecção de objetos. Esse padrão consiste de um conjunto de dados de detecção, segmentação, detecção de pontos-chave e legendas de objetos em grande escala que possui um conjunto de imagens de 328K criado pela Microsoft. Dessa forma, a possibilidade de interação com a base proposta aumenta e facilita o uso em diferentes cenários de aplicação.

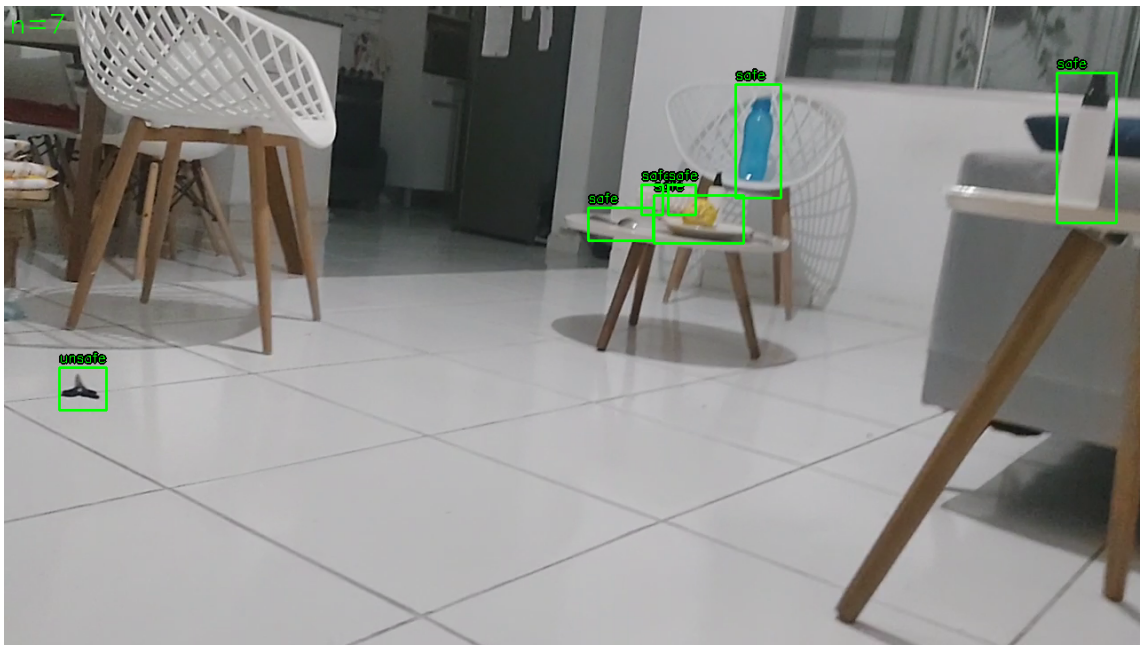
Figura 4 - Anotação gerada no formato COCO



Fonte: O autor

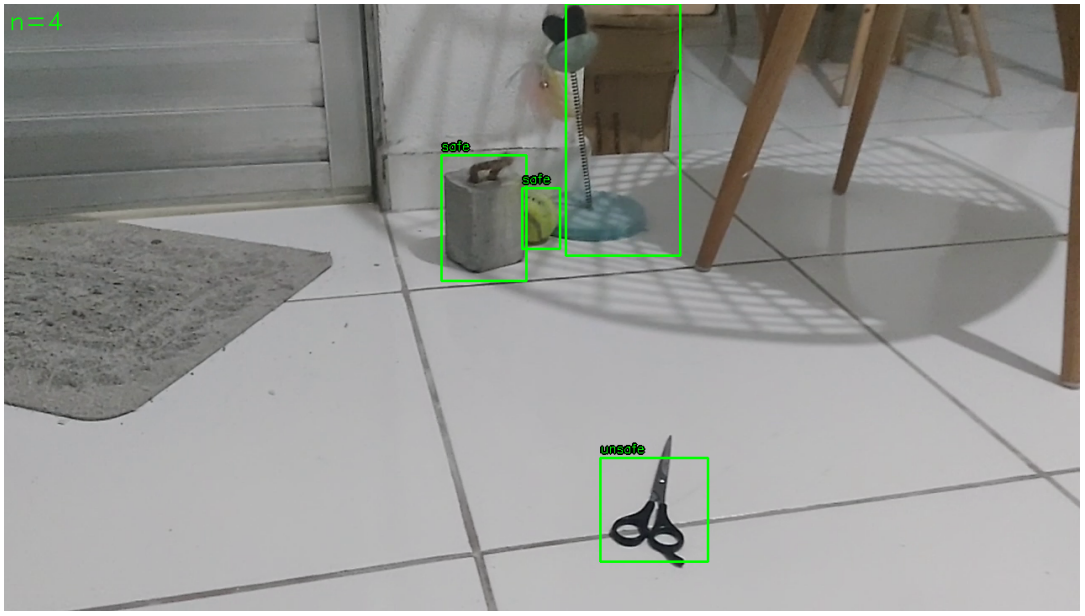
Abaixo está a captura de tela de alguns quadros de um vídeo utilizado após anotação na plataforma Darklabel. As caixas delimitadoras definidas manualmente buscam preencher todo o espaço de um ou múltiplos objetos em foco ao passo em que se percorre o ambiente. Ao encontrá-los, criar-se-ia uma nova linha identificando tal objeto no arquivo .xml para a imagem corrente, e, dessa forma, a anotação de cada vídeo compartilhado é feita.

Figura 5 - Identificação de sete objetos na cena, dentre eles seis objetos seguros e um periculoso



Fonte: O Autor

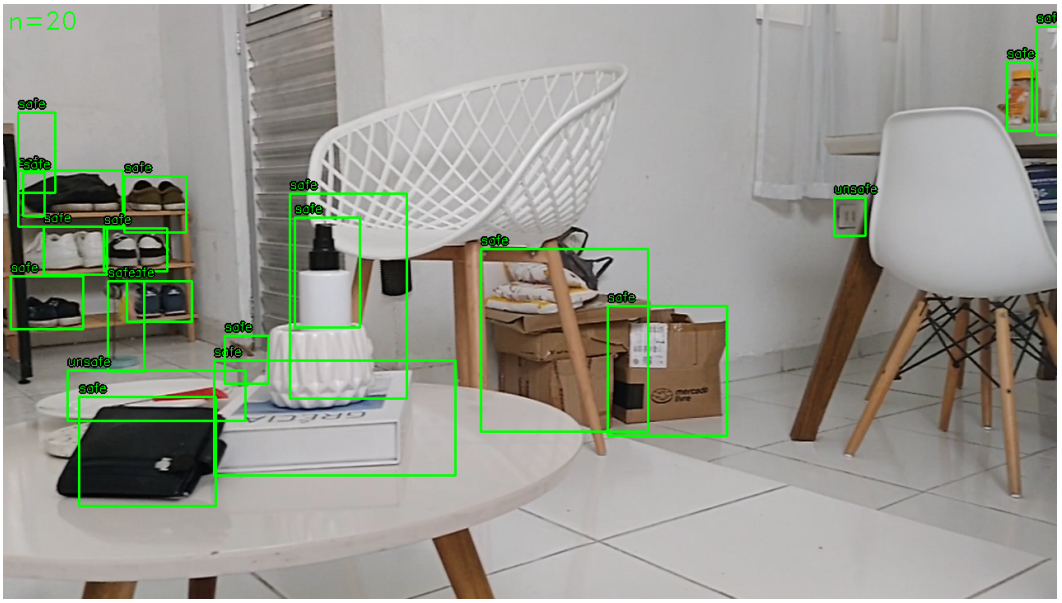
Figura 6 - Identificação automática de quatro objetos na cena, sendo um objeto periculoso e três seguros



Fonte: O Autor

Em interações posteriores a de uma definição manual de caixas delimitadoras nos objetos de uma imagem, é possível utilizar a previsão automática do Darklabel para identificar os conteúdos. Ainda sim, tal suporte não se mostrou tão preciso em diferentes casos. Por conseguinte, era necessário delimitar os objetos pelas caixas classificadores a cada 20 quadros, em média. Ainda sim, considerando que um objeto permaneceria aproximadamente 100 quadros por posição em vídeo, é possível afirmar que o modelo treinado conseguirá construir o objeto tal qual a sua classificação corretamente, dado todo o espaço amostral.

Figura 7 - Identificação manual de 20 objetos na cena, sendo dois objetos periculosos e dezessete seguros



Fonte: O Autor

A própria ferramenta *DarkLabel* ressalta na imagem a quantidade de caixas delimitadoras de conteúdo que estão sendo usadas a cada interação. Dessa forma, ao verificar que um conteúdo continua sendo classificado e não está contido explicitamente na imagem atual, conclui-se que que o objeto saiu de cena e aquela classificação não deverá mais existir. Os objetos presentes na imagem, consequentemente, conseguirão ser identificados com todos seus atributos a partir das tags de object encontradas em cada xml gerado.

5.4 Ferramentas de desenvolvimento

Foram utilizadas como ferramentas durante o desenvolvimento do trabalho *Google Colab* para processamento de conteúdo utilizando a linguagem de programação Python, além disso foram utilizadas as bibliotecas do *Scikit Learn[86], Tensorflow[87]*, *OpenCV[82]*. Uma parte do código ficou disponível num repositório remoto do Github, e a outra ficou disponível em um ambiente de desenvolvimento do *Google Colab*. Para a confecção do presente documento, foram utilizadas a ferramenta *Git* para manusear os repositórios e um computador Dell Latitude i7 Pro com 16gb de RAM com um processador gráfico integrado e o sistema operacional utilizado foi o Linux Ubuntu 20.04 LTS.

6 Resultados

6.1 Base

Ao final da coleta de dados, foram recebidas 6 respostas, contendo o total de 140 vídeos, onde 66,7% dos dados foram excluídos após análise manual criteriosa de sua construção pelas métricas objetivas dissertadas na fase de coleta de dados.

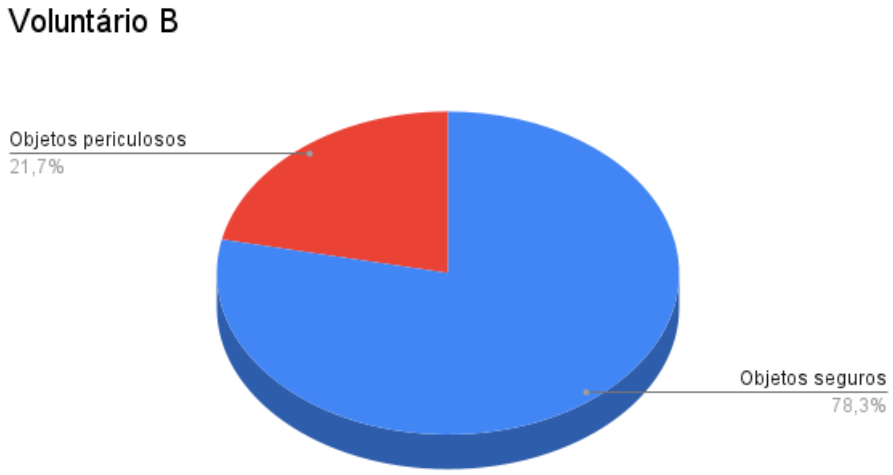
Os principais motivos que levaram a exclusão do conteúdo foram:

- Posicionamento errado da câmera além do estipulado nas necessidades iniciais
- Objetos periculosos diferentes do escolhido
- Velocidade exacerbada do autor do vídeo, inibindo clareza na anotação

Após a exclusão dos conteúdos que não estavam de acordo com o estipulado para criação da base, dois conjuntos de dados foram utilizados para construção da base.

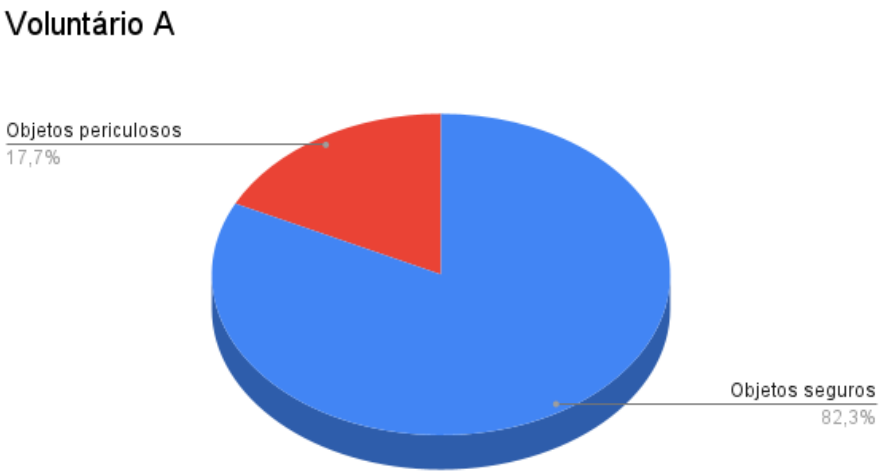
Considerando que foram classificados 119204 objetos pelos vídeos do voluntário A e 316549 objetos pelos vídeos do voluntário B, a distribuição do conteúdo da base a partir dos dados do Voluntário A e do Voluntário B ficou da seguinte forma:

Figura 8 - Apuração da anotação feita para os dados enviados pelo voluntário B



Fonte: O Autor

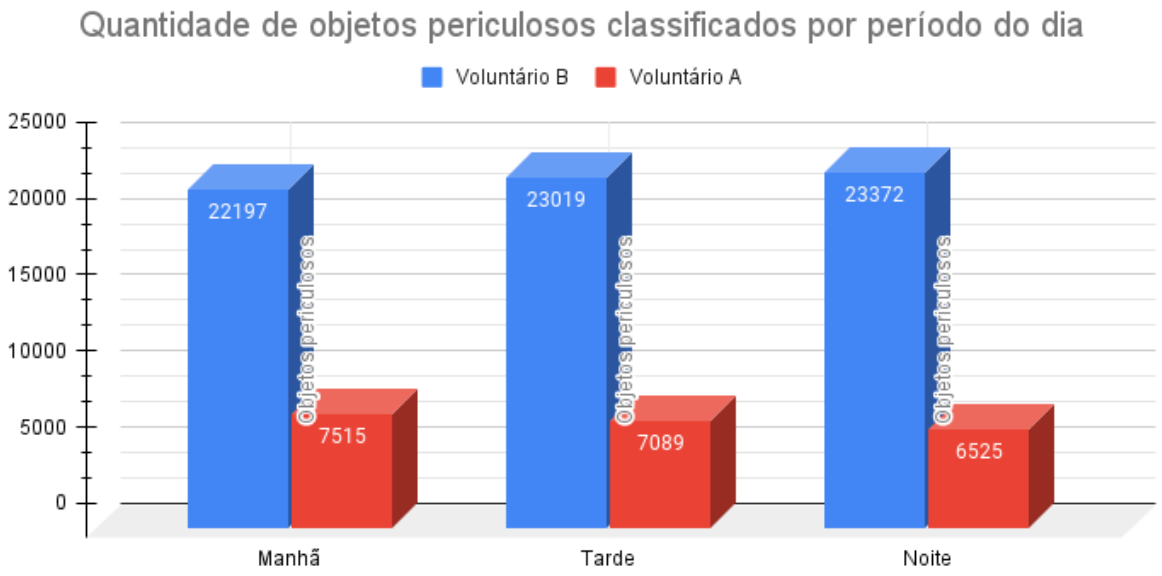
Figura 9 - Apuração da anotação feita para os dados enviados pelo voluntário A



Fonte: O Autor

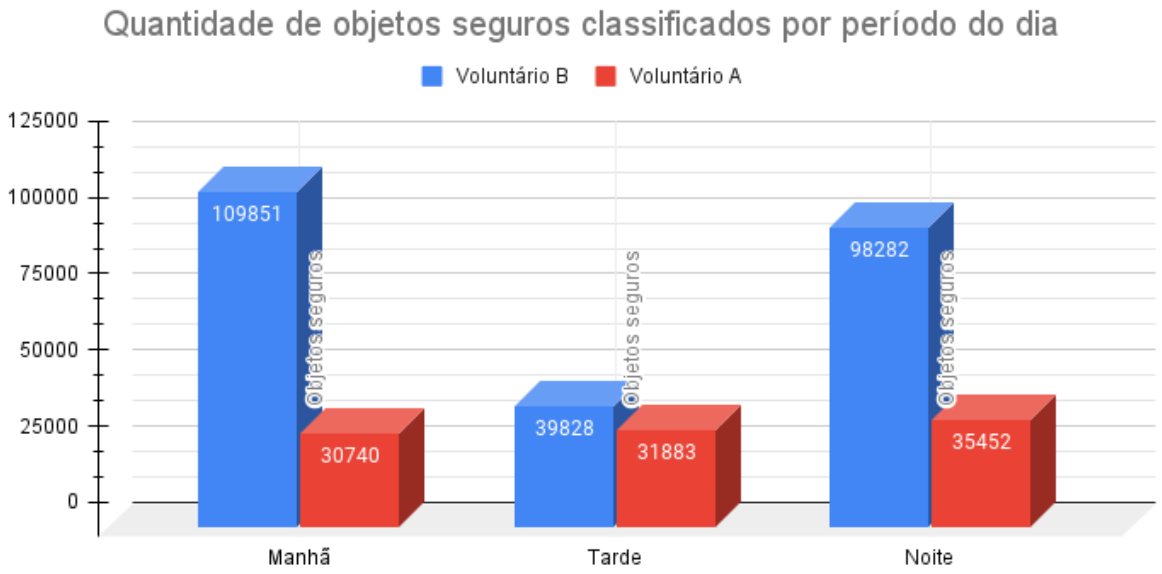
Abaixo é ilustrada uma relação comparativa entre a entrada dos dados dos dois usuários, mostrando como está a distribuição das anotações dada ambas classes utilizadas na criação da base.

Figura 10 - Distribuição por horários das incidências de objetos perigosos por voluntário



Fonte: O Autor

Figura 11 -Distribuição por horários das incidências de objetos perigosos por voluntário



Fonte: O Autor

6.2 Classificação

Com o objetivo de buscar uma validação da base proposta, foi utilizado uma arquitetura end-to-end para almejar resultados. O transformador DETR [88] consiste de forma simplificada em três pontos.

- As camadas CNN são usadas para extrair características da imagem,
- Possui estrutura codificador-decodificador no Transformador
- Lida com a função de perda de um conjunto, executando por uma correspondência bipartida entre objetos previstos e reais

Como essas operações são realizadas em paralelo a partir de arquiteturas de transformadores, é possível obter um modelo muito mais rápido. E com a utilização da GPU disponível no Google Colab ao invés do processamento em CPU, há também uma diminuição do tempo de processamento.

A partir da explicação proposta no Capítulo 3 acerca das escolhas necessárias para construção de um algoritmo de aprendizagem de máquina como a CNN, foi considerado os seguintes valores para as configurações desse transformador.

Tabela 5 - Argumentos necessários para inicializar o DETR

Argumentos de Inicialização
Taxa de aprendizagem base = 2e-4
Nome do backbone = ["backbone.0"]
Taxa de aprendizagem do backbone = 2e-5
Camadas de projeção linear que usam taxas menores = ['reference points', 'sampling_offsets']
Valor da taxa = 0.1
Tamanho do batch = 2
Queda de peso = 1e-4
Número máximo de *Epochs* = 50
Queda da taxa de aprendizagem = 40
Excluir epochs = Nenhum
Gradiente de recorte = 0.1
Uso de gradiente estocástico = Sim

Fonte: O autor

Tabela 6 - Variantes utilizadas pelo DETR referente aos estágios e refinamento do processo

Variantes do *Deformable DETR*
Refinar caixa delimitadora = Falso
Estágios múltiplos = Falso

Fonte: O autor

Tabela 7 - Valor ao ser utilizado para como o modelo deve se portar

Parâmetros do modelo
Pesos congelados = Nenhum

Fonte: O autor

Tabela 8 - Configuração do Backbone a ser conectado

Backbone
Tipo de backbone = 'resnet50'
Dilatação = Sim
Tipo de incorporação posicional para usar em cima dos recursos de vídeo = 'sine'

Escala de incorporação de posição = 2 * np.pi
Níveis = 4

Fonte: O autor

Tabela 9 - Valores principais do Transformador

Transformador
Número de camadas de codificação no transformador = 6
Número de camadas de decodificação no transformador = 6
Tamanho intermediário das camadas *feedforward* nos blocos do transformador = 1024
Tamanho dos encaixes (dimensão do transformador) = 256
Dropout aplicado no transformador = 0.1
Número de cabeças de atenção dentro das atenções do transformador = 8
Número de *queries* de ação = 300
Número de pontos amostrados por cabeça para atenção deformável no decodificador = 4
Número de pontos amostrados por cabeça para atenção deformável no codificador = 4

Fonte: O autor

Tabela 10 - Escolha da utilização de máscaras para segmentação

Segmentação
Usos de máscaras = Falso

Fonte: O autor

Tabela 11 - Configuração de perda

Perda
Habilitar perdas de decodificação auxiliares (perda em cada camada) = Falso

Fonte: O autor

Tabela 12 - Definições acerca da identificação do conteúdo em cena

Parâmetros para o *match* de objetos
Definir peso de custo - Coeficiente de classe = 2
Definir peso de custo - Caixas delimitadoras = 5
Definir peso de custo - Coeficiente de Segmento IoU = 2

Fonte: O autor

Tabela 13 - Seleção dos coeficientes de perda

Coeficientes de perda
Coeficiente de perda para máscara = 1
Coeficiente de perda para Sorensen-Dice = 1
Coeficiente de perda por classe = 2
Coeficiente de perda para caixas delimitadoras = 5
Coeficiente de perda para IoU = 2
Coeficiente de perda para perda focal = 0.25

Fonte: O autor

Tabela 14 - Configuração necessária para acesso ao banco de anotações desenvolvido

Parâmetros do Dataset
Tipo de arquivo de anotação = 'coco'
Caminho para o *dataset* = './data/hod_database'
Remoção de dificuldades = Sim
Quantidade de classes classificadas = 2
Dispositivo utilizado para o processamento = 'cuda'
Seed = 42
Epoch = 0
Avaliação = Sim
Número de trabalhadores = 2
Cache = Falso

Fonte: O autor

Com a configuração acima referente aos pontos discutidos no capítulo 2, 3 e 4 sendo dispostas em uma classe inicializadora, é possível dizer que a base hoje já está em uso no cenário almejado, promovendo suas primeiras interações.

7. Conclusão e Trabalhos futuros

Neste capítulo são apresentadas as conclusões do presente trabalho e as recomendações para a continuidade dos trabalhos nesta área de estudo.

7.1 Conclusões

Foi possível gerar uma base de dados que possui em sua totalidade 28,10 GB de dados gerados por 36 vídeos que analisaram diferentes cenários domésticos com objetivo de identificar objetos perigosos, sendo eles tomadas, facas e tesouras.

A utilização de implementações open source como *Darklabel* e *DETR Transformer* implicou no crescimento do tempo para cada fase da execução do projeto, uma vez que algumas dificuldades foram encontradas durante o processo de construção da base.

O *Darklabel* possui uma limitação clara na forma de anotação dos objetos em cena. Sendo inibida a capacidade de apagar a identificação de um objeto em um dos frames, era necessário reiniciar a anotação do vídeo para que houvesse prosseguimento da tarefa de anotação de cada vídeo.

A renderização dos conteúdos anotados e posterior armazenagem desses dados era impactada diretamente pelo hardware utilizado nas anotações, levando cerca de 5 a 10 minutos para salvar todos os frames de cada um dos vídeos analisados. Acerca da anotação dos arquivos, cada um dos vídeos levava cerca de 40 minutos para ser classificado. Para a geração do first run promovido pelo transformador, esse valor temporal consegue aumentar expressivamente, levando mais de 3 dias para promover resultados referentes à base de dados.

Uma vez que um *conjunto de anotações* contenha os rótulos associados aos arquivos de origem enviados em um conjunto de dados, esse conjunto de anotações está associado a um tipo de dados e um objetivo. Ao criar um conjunto de dados e exportá-lo, dá-se a oportunidade de outros desenvolvedores utilizarem da construção para aperfeiçoar ou manusear suas necessidades acadêmicas e profissionais. que contém rótulos, um conjunto de anotações é criado e atribuído o objetivo selecionado no momento da criação.

7.2 Trabalhos futuros

Como foi dito anteriormente, a construção da base de dados foi uma etapa que levou um tempo considerável.

Em função da indisponibilidade de geração de algumas informações e do tempo para conclusão deste trabalho, recomenda-se para trabalhos futuros a incorporação ao presente modelo de bases de dados mais diversas com um maior número de imagens e que possuam outras rotações e formas de se visualizar o objeto.

A solicitação de novas entradas de dados podem gerar modificações significativas nos resultados propostos pelo modelo final. Dessa forma, indica-se em sequência que seja feito uma nova arrecadação de vídeos para que novos arquivos de anotações sejam gerados e consequentemente auxiliem no processo de aprendizagem do conhecimento.

Também é possível utilizar outros modelos, backbones, transformadores e ferramentas para anotação de vídeo com o objetivo de identificar os prós e os contras do uso desses utilitários na construção e validação de uma base de anotações do zero.

É proposto também a utilização dos resultados gerados a partir da base desenvolvida em classificações de outros dados no mesmo contexto, ou seja, bases de testes mais randômicas, com o objetivo de verificar a acurácia dos valores resultantes.

Por fim, sugere-se também a incorporação de uma análise comparativa entre os diferentes modelos presentes no estado da arte deste tópico de VC, de forma a validar qualitativamente as saídas geradas e com isso definir as melhores escolhas de configuração a serem utilizadas ao se manusear tais ferramentas discutidas no presente desenvolvimento

APÊNDICE A - INTRODUÇÃO DO FORMULÁRIO

Olá!

Neste formulário busco diferentes entradas de gravações para auxiliar na modelagem de um classificador utilizado no meu trabalho de conclusão de curso. A seguir, serão dadas algumas instruções importantes caso você decida colaborar nesse projeto.

Para começar, as gravações deverão ser feitas em algum cômodo de sua própria residência a partir de uma câmera celular.

Para a pesquisa, serão utilizados alguns materiais. Sendo eles: tomadas (independente do formato e construção), facas (independente do tamanho, ou de onde ela é utilizada) e tesouras (independente de ser profissional, com ponta, tamanho ou cor).

Por fim, estima-se que a geração de dados por cada indivíduo fique em torno de 30 minutos, somando todas as necessidades da pesquisa, incluindo as variações de horário para gravações tal qual construção do formato ideal. As informações de toda a construção dos dados estão detalhadas na página referente ao anexo dos dados individuais.

Qualquer dúvida, favor entrar em contato a partir do email rjmr@ecomp.poli.br

Fonte: O Autor

APÊNDICE B - TERMO DE CONCORDÂNCIA

Declaração do Autor

Eu, Richard Jeremias Martins Rocha, discente da Graduação em Engenharia da Computação da Universidade de Pernambuco, no âmbito do projeto de pesquisa do trabalho de conclusão de curso intitulado provisoriamente de Construindo uma base para validar aspectos de semântica em robótica assistiva, sob orientação do professor Carmelo José Albanez Bastos Filho, comprometo-me com a utilização dos dados contidos nesse formulário, a fim de obtenção dos objetivos previstos.

Comprometo-me a manter a confidencialidade dos dados coletados aqui, bem como com a privacidade de seus conteúdos, ou seja, os vídeos e conteúdos aqui anexados.

Declaro entender que é minha a responsabilidade de cuidar da integridade das informações e de garantir a confidencialidade dos dados e a privacidade dos indivíduos que terão suas informações acessadas.

Também é minha a responsabilidade de não repassar os dados coletados ou o banco de dados em sua íntegra, ou parte dele, às pessoas não envolvidas na equipe da pesquisa.

Por fim, comprometo-me com a guarda, cuidado e utilização das informações apenas para cumprimento dos objetivos previstos nesta pesquisa aqui referida.

Esclareço ainda que os dados coletados farão parte dos estudos do aluno Antonio Victor Alencar Lundgren, discente de Doutorado em Engenharia de Computação da Universidade de Pernambuco, sob orientação do professor Carmelo José Albanez Bastos Filho.

Recife, 30 de julho de 2022.

Fonte: O Autor

Referências

[1] BERSCH, Rita. Introdução à tecnologia assistiva. Porto Alegre: CEDI, v. 21, 2008.

[2] COOK, Albert M.; POLGAR, Janice Miller. Assistive technologies e-book: principles and practice. Elsevier Health Sciences, 2014.

[3] BERSCH, Rita; SARTORETTO, Mara L. O que é Tecnologia Assistiva? Assistiva, 2022. Disponível em: <https://www.assistiva.com.br/tassistiva.html>. Acesso em: 10/08/2022.

[4] SHAPIRO, Linda G. et al. Computer vision. New Jersey: Prentice Hall, 2001.

[5] MINARI, Gustavo. Inteligência artificial e aprendizagem de máquina são a mesma coisa? Entenda. Canaltech, 2019. Disponível em: <https://canaltech.com.br/inovacao/inteligencia-artificial-e-aprendizagem-de-maquina-sao-a-mesma-coisa-entenda-195558/> Acesso em: 10/08/2022.

[6] TIWARI, Rohit Kumar; VERMA, Gyanendra K. A computer vision based framework for visual gun detection using harris interest point detector. Procedia Computer Science, v. 54, p. 703-712, 2015.

[7] VAROTSIS, Andreas. Automating Knife Classification with Machine Learning. Disponível em: <https://andreas-varotsis.medium.com/automating-knife-classification-with-machine-learning-3767d93d2789> Acesso em: 10/08/2022.

[8] BERGELES, Christos; YANG, Guang-Zhong. From passive tool holders to microsurgeons: safer, smaller, smarter surgical robots. IEEE Transactions on Biomedical Engineering, v. 61, n. 5, p. 1565-1576, 2013.

[9] GOMBOLAY, Matthew et al. Robotic assistance in the coordination of patient care. The International Journal of Robotics Research, v. 37, n. 10, p. 1300-1316, 2018.

[10] VERCELLI, Alessandro et al. Robots in elderly care. DigitCult-Scientific Journal on Digital Cultures, v. 2, n. 2, p. 37-50, 2018.

[11] CROSSMAN, Molly K.; KAZDIN, Alan E.; KITT, Elizabeth R. The influence of a socially assistive robot on mood, anxiety, and arousal in children. Professional Psychology: Research and Practice, v. 49, n. 1, p. 48, 2018.

[12] MANTI, Mariangela et al. Soft assistive robot for personal care of elderly people. In: 2016 6th IEEE international conference on biomedical robotics and biomechatronics (BioRob). Ieee, 2016. p. 833-838

[13] ZHANG, Xiangyu et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. p. 6848-6856.

[14] ZHANG, Lei; WANG, Shuai; LIU, Bing. Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v. 8, n. 4, p. e1253, 2018.

[15] YOUNG, Tom et al. Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine, v. 13, n. 3, p. 55-75, 2018

[16] LUNDGREN, Antonio. Systematic Review of Computer Vision Semantic Analysis in Socially Assistive Robotics. AI, v. 3, n. 1, p. 229-249, 2022.

[17] GIRSHICK, R; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014

[18] HAFIZ, Abdul Mueed; BHAT, Ghulam Mohiuddin. A survey on instance segmentation: state of the art. International journal of multimedia information retrieval, v. 9, n. 3, p. 171-189, 2020.

[19] LEMPITSKY, V.; KOHLI, P.; ROTHER , C.; SHARP, T. "Image segmentation with a bounding box prior," 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 277-284, doi: 10.1109/ICCV.2009.5459262.

[20] VIEIRA, Leonardo Lourenço. Desenvolvimento de interfaces homem-máquina de alta performance. 2018.

[21] ZHU, Z. et al. Wearable navigation assistance for the vision-impaired. [S.l.]: Google Patents, 2014. US Patent App. 14/141,742.

[22] JIA, P. et al. Head gesture recognition for hands-free control of an intelligent wheelchair. Industrial Robot: An International Journal, 2007. Emerald Group Publishing Limited, v. 34, n. 1, p. 60–68, 2007. Citado na página 26.

[23] VINCZE, M.; WACHSMUTH, S.; SAGERER, G. Perception and computer vision. In K. Frankish & W. Ramsey (Eds.), The Cambridge Handbook of Artificial Intelligence (pp. 168-190). Cambridge: Cambridge University Press. 2014.

[24] BAMBA, Yoshiko et al. Object and anatomical feature recognition in surgical video images based on a convolutional neural network. International Journal of Computer Assisted Radiology and Surgery, v. 16, n. 11, p. 2045-2054, 2021.

[25] GHISI, Gabriela Caroline, JÚNIOR, Gastão Dias, FACHINI, Janaína Sortica, SANTOS, Tatiana Coutinho. Perfil epidemiológico das internações por acidentes domiciliares em um hospital pediátrico da região sul do Brasil, Arq. Catarin Med, 2018.

[26] BEZERRA, Maria Augusta Rocha et al. Acidentes domésticos em crianças: concepções práticas dos agentes comunitários de saúde. Cogitare Enferm, Piauí, v. 4, n. 19, p.776-784, out. 2014.

[27] SILVA, Manalde Ferreira da et al. Fatores determinantes para a ocorrência de acidentes domésticos na primeira infância. J. Hum. Growth Dev. 2017, vol.27, n.1 , pp. 10-18 . Sociedade Brasileira de Pediatria. Queimaduras. São Paulo: Sociedade Brasileira de Pediatria; 2014.

[28] GARCIA, Ana Cristina. Ética e inteligência artificial. Computação Brasil, n. 43, p. 14-22, 2020.

[29] INFORMATION AGE. Disponível em: <https://www.information-age.com/success-artificial-intelligence-data-123471607> Acesso em: 18/06/2022.

[30] UPADHYAY, Ashwani Kumar; KHANDELWAL, Komal. Artificial intelligence-based training learning from application. Development and Learning in Organizations: An International Journal, 2019.

[31] HAYKIN, Simon. Redes neurais: princípios e prática. Bookman Editora, 2001.

[32] LUDERMIR, Teresa Bernarda. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. Estudos Avançados, v. 35, p. 85-94, 2021.

[33] TACHIBANA, Wilson Kendy; Orlandi, Veridiana De Fátima. Redes Neurais Artificiais: Uma Contribuição Ao Processo De Decisões Financeiras. In: Anais do Congresso Brasileiro de Custos-ABC. 1998.

[34] TESAURO, Gerald et al. Temporal difference learning and TD-Gammon. Communications of the ACM, v. 38, n. 3, p. 58-68, 1995.

[35] ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, v. 65, n. 6, p. 386, 1958.

[36] Expert IA, 2020. Importância do bias nas redes neurais. Disponível em <https://iaexpert.academy/2020/09/28/importancia-do-bias-nas-redes-neurais/>. Acesso em: 02/10/2022.

[37] MCCULLOCK, W. S.; PITTS, W. A logical calculus of ideas immanent in nervous activity. archive copy of 27 november 2007 on wayback machine. Avtomaty [Automated Devices] Moscow, Inostr. Lit. publ, p. 363-384, 1956.

[38] HAYKIN, Simon. Neural networks and learning machines, 3/E. Pearson Education India, 2009.

[39] SELAU, Lisiane Priscila Roldão. Redes neurais artificiais no contexto estatístico. 2000.

[40] HAIR, Joseph F. et al. Multivariate data analysis . Uppersaddle River. Multivariate Data Analysis (5th ed) Upper Saddle River, v. 5, n. 3, p. 207-219, 1998.

[41] TONSIG, Sérgio Luiz. Redes Neurais Artificiais Multicamadas e o Algoritmo de Backpropagation. Campinas, 2000. Disponível em: http://209.123.181.8/~archives/tutoriais/1243.zip. Acesso em 02/09/2022.

[42] CARVALHO, A. P. de L. F. Redes Neurais Artificiais. Departamento de Ciência da Computação. Disponível em: <http://icmc.sc.usp.br/~andre/neural2.html>. Acesso em: 03/10/2022.

[43] DATA ANALYSIS WITH PYTHON - IMAGE PROCESSING. MOOC, 2022. Disponível em: <https://dap-21.mooc.fi/week-3/image_processing>. Acesso em: 07/10/2022.

[44] PARKER, Jim R. Algorithms for image processing and computer vision. John Wiley & Sons, 2010.

[45] LECUN, Yann et al. “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2323, 1998

[46] FEI-FEI, Li; FERGUS, Robert; PERONA, Pietro. “One-shot learning of object categories,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 594–611, 2006.

[47] KRIZHEVSKY, Alex et al. Learning multiple layers of features from tiny images, 2009.

[49] ARSENOV, Andrey et al. Evolution of Convolutional Neural Network Architecture in Image Classification Problems. In: ITS. 2018. p. 35-45.

[48] PEREIRA, Roberto Matheus Pinheiro. Estudo do desempenho de redes neurais convolucionais aplicada ao reconhecimento de símbolos musicais, glaucoma e texto. 2017.

[50] CARVALHO, A. P. de L. F. Perceptron Multicamadas. Departamento de Ciência da Computação. Disponível em: <http://icmc.sc.usp.br/~andre/neural2.html>. Acesso em: 11/10/2022.

[51] EIDINGER, Eran; ENBAR, Roee; HASSNER, Tal. “Age and gender estimation of unfiltered faces,” IEEE Transactions on Information Forensics and Security, vol. 9, no. 12, pp. 2170–2179, 2014.

[52] WANG, Xiaosong et al. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 2097-2106.

[53] SEFF, Ari et al. Leveraging mid-level semantic boundary cues for automated lymph node detection. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, 2015. p. 53-61.

[54] VOULODIMOS, Athanasios et al. A dataset for workflow recognition in industrial scenes. In: 2011 18th IEEE International Conference on Image Processing. IEEE, 2011. p. 3249-3252.

[55] VOULODIMOS, Athanasios et al. A threefold dataset for activity and workflow recognition in complex industrial environments. IEEE MultiMedia, v. 19, n. 03, p. 42-52, 2012.

[56] KOSMOPOULOS, Dimitrios I.; VOULODIMOS, Athanasios S.; DOULAMIS, Anastasios D. A system for multicamera task recognition and summarization for structured environments. IEEE Transactions on Industrial Informatics, v. 9, n. 1, p. 161-171, 2012.

[57] ABU-EL-HAIJA, Sami et al. Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675, 2016.

[58] HINTON, Geoffrey E.; OSINDERO, Simon; TEH, Yee-Whye. A fast learning algorithm for deep belief nets. Neural computation, v. 18, n. 7, p. 1527-1554, 2006.

[59] KARHUNEN, Juha; RAIKO, Tapani; CHO, KyungHyun. Unsupervised deep learning: A short review. Advances in independent component analysis and learning machines, p. 125-142, 2015.

[60] SARKER, Iqbal H. Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective. SN Computer Science, v. 2, n. 3, p. 1-16, 2021.

[61] CARNEIRO, Herman Anthony; MYLONAKIS, Eleftherios. Google trends: a web-based tool for real-time surveillance of disease outbreaks. Clinical infectious diseases, v. 49, n. 10, p. 1557-1564, 2009.

[62] XIN, Yang et al. Machine learning and deep learning methods for cybersecurity. Ieee access, v. 6, p. 35365-35381, 2018.

[63] UIJLINGS, Jasper RR et al. Selective search for object recognition. International journal of computer vision, v. 104, n. 2, p. 154-171, 2013.

[64] GIRSHICK, Ross. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2015. p. 1440-1448.

[65] REN, Shaoqing et al. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, v. 28, 2015.

[66] HOSANG, Jan; BENENSON, Rodrigo; SCHIELE, Bernt. How good are detection proposals, really?. arXiv preprint arXiv:1406.6962, 2014.

[67] HARIHARAN, Bharath et al. Simultaneous detection and segmentation. In: European conference on computer vision. Springer, Cham, 2014. p. 297-312.

[68] DONG, Jian et al. Towards unified object detection and semantic segmentation. In: European Conference on Computer Vision. Springer, Cham, 2014. p. 299-314.

[69] ZHU, Yukun et al. segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. p. 4703-4711.

[70] LIU, Jiamin et al. Colitis detection on abdominal CT scans by rich feature hierarchies. In: Medical Imaging 2016: Computer-Aided Diagnosis. SPIE, 2016. p. 423-429.

[71] LUO, Gongning et al. A deep learning network for right ventricle segmentation in short-axis MRI. In: 2016 Computing in Cardiology Conference (CinC). IEEE, 2016. p. 485-488.

[72] OUYANG, Wanli et al. DeepID-Net: Object detection with deformable part based convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 39, n. 7, p. 1320-1334, 2016.

[73] DIBA, Ali et al. Weakly supervised cascaded convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 914-922.

[74] CHEN, Tao; LU, Shijian; FAN, Jiayuan. S-CNN: Subcategory-aware convolutional networks for object detection. IEEE transactions on pattern analysis and machine intelligence, v. 40, n. 10, p. 2522-2528, 2017.

[75] DIAO, Wenhui et al. Efficient saliency-based object detection in remote sensing images using deep belief networks. IEEE Geoscience and Remote Sensing Letters, v. 13, n. 2, p. 137-141, 2016.

[76] NAIR, Vinod; HINTON, Geoffrey E. 3D object recognition with deep belief nets. Advances in neural information processing systems, v. 22, 2009.

[77] DOULAMIS, Nikolaos; DOULAMIS, Anastasios. Fast and adaptive deep fusion learning for detecting visual objects. In: European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012. p. 345-354.

[78] DOULAMIS, Nikolaos; DOULAMIS, Anastasios. Semi-supervised deep learning for object tracking and classification. In: 2014 IEEE international conference on image processing (ICIP). IEEE, 2014. p. 848-852.

[79] SHIN, Hoo-Chang et al. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data. IEEE transactions on pattern analysis and machine intelligence, v. 35, n. 8, p. 1930-1943, 2012.

[80] LI, Jia; XIA, Changqun; CHEN, Xiaowu. A benchmark dataset and saliency-guided stacked autoencoders for video-based salient object detection. IEEE Transactions on Image Processing, v. 27, n. 1, p. 349-364, 2017.

[81] SOLEM, Jan Erik. Programming Computer Vision with Python: Tools and algorithms for analyzing images. " O'Reilly Media, Inc.", 2012.

[82] CHANDAN, G. et al. Real time object detection and tracking using Deep Learning and OpenCV. In: 2018 International Conference on inventive research in computing applications (ICIRCA). IEEE, 2018. p. 1305-1308.

[83] DARKLABEL, VIDEO/IMAGE LABELING AND ANNOTATION TOOL. Github, 2020. Disponível em < https://github.com/darkpgmr/DarkLabel>. Acesso em: 10/08/2022.

[84] ALMEIDA, Samuel Sousa. Identificação de features de campo com OpenCV para uso em robô NAO. 2020.

[85] LIN, Tsung-Yi et al. Microsoft coco: Common objects in context. In: European conference on computer vision. Springer, Cham, 2014. p. 740-755.

[86] PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, v. 12, p. 2825-2830, 2011.

[87] ABADI, Martín et al. TensorFlow: a system for Large-Scale. machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16). 2016. p. 265-283.

[88] FUNDAMENTAL Vision. Deformable DETR Deformable - Transformers for End-to-End Object Detection. Disponível em: https://github.com/fundamentalvision/Deformable-DETR/. Acesso em: 04/08/2022.