

**UNIVERSIDADE PRESBITERIANA MACKENZIE
FACULDADE DE COMPUTAÇÃO E INFORMÁTICA
CURSO DE TECNOLOGIA EM CIÊNCIAS DE DADOS**

**IZAQUE NAZARENO DE MELO SOUZA
DOUGLAS PEREIRA DE ARAÚJO**

**DETECÇÃO E CONTAGEM DE OBJETOS ESPECÍFICOS EM MÚLTIPLAS
IMAGENS**

**SÃO PAULO
2023**

IZAQUE NAZARENO DE MELO SOUZA

DOUGLAS PEREIRA DE ARAÚJO

**DETECÇÃO E CONTAGEM DE OBJETOS ESPECÍFICOS EM MÚLTIPLAS
IMAGENS**

Trabalho apresentado como requisito
para obtenção de nota na disciplina
Projeto Aplicado II do curso de
Tecnologia em Ciências de Dados da
Universidade Presbiteriana Mackenzie,
ministrada pelo professor Dr. Anderson
Adaime de Borba

SÃO PAULO

2023

SUMÁRIO

1 INTRODUÇÃO	3
2 DESENVOLVIMENTO	3
2.1 DEFINIÇÃO DE BIBLIOTECAS UTILIZADAS	4
2.2 DEFINIÇÃO DA BASE DE DADOS E ANÁLISE EXPLORATÓRIA	5
2.4 MÉTODOS ANALÍTICOS UTILIZADOS	7
2.3 MÉTRICA DE AVALIAÇÃO	8
3 REPOSITÓRIO PARA DISPONIBILIZAÇÃO DOS DADOS.....	8
4 CRONOGRAMA	8
BIBLIOGRAFIA	10

1 INTRODUÇÃO

A detecção de objetos em imagens é uma técnica de processamento de imagem que consiste em localizar objetos específicos dentro de uma imagem digital (ZHOU, 2021). Essa técnica é frequentemente usada em aplicações de visão computacional, como segurança, robótica, reconhecimento facial, laudos médicos, entre outras (KARTHI, M. et al, 2021; JIANG, 2022).

Essa técnica geralmente envolve a criação de um modelo que é treinado para reconhecer padrões e características de objetos específicos em uma imagem. Esse modelo pode usar técnicas de aprendizado de máquina, como redes neurais convolucionais (CNNs), para identificar objetos em uma imagem (HUANG, Zhanchao et al, 2020). Uma vez que o modelo é treinado, ele pode ser usado para detectar objetos em novas imagens. A detecção de objetos geralmente envolve a aplicação do modelo a uma imagem e a identificação dos objetos detectados por meio de caixas delimitadoras ao redor de cada objeto.

Dessa forma, neste trabalho será realizado a detecção e contagem de objetos específicos de forma precisa, robusta, rápida e capaz de identificar a maioria dos objetos presentes na imagem, minimizando a ocorrência de falsos positivos. Como objetos de identificação, o algoritmo deverá ser capaz de identificar gatos em diversos ambientes e imagens junto com outros animais (cachorro). As imagens onde os objetos serão identificados e contados serão imagens pessoais dos autores de ambientes internos e externo. Para

2 DESENVOLVIMENTO

Figura 01 – Breve fluxograma do processo de detecção de objetos utilizado neste trabalho.



Acima exemplificamos o processo de detecção de objetos adotado neste trabalho, nos próximos tópico, descrevemos com mais detalhe cada etapa.

2.1 DEFINIÇÃO DE BIBLIOTECAS UTILIZADAS

Para o desenvolvimento da atividade, serão utilizadas as seguintes bibliotecas:

- **Matplotlib:** é uma das bibliotecas mais populares para visualização de dados em Python e fornece uma ampla gama de recursos para criação de gráficos e visualizações. Também será utilizado o seu módulo "pyplot" que contém uma série de funções para criar e personalizar gráficos, como histogramas, boxplots, gráficos de linha, gráficos de dispersão, entre outros;
- **Pandas:** Utilizada para análise e manipulação de dados em Python. Ela fornece estruturas de dados de alto nível, fáceis de usar e eficientes em termos de desempenho, especialmente para tarefas como a leitura, limpeza, transformação e análise de dados tabulares.
- **TensorFlow:** biblioteca utilizada para aprendizado de máquina e deep learning em Python. Fornece uma plataforma completa para criação e treinamento de modelos de aprendizado de máquina e deep learning, desde modelos simples até modelos complexos de redes neurais profundas.
- **OpenCV:** *Open Source Computer Vision* (OpenCV) é uma biblioteca de código aberto amplamente utilizada para processamento de imagens e vídeos. Ele fornece uma ampla variedade de funções para capturar, manipular e analisar imagens e vídeos em tempo real, como reconhecimento facial, detecção de objetos, rastreamento de movimento, entre outros.
- **YOLO:** *You Only Look Once* (YOLO) é um algoritmo de detecção de objetos em imagens. O algoritmo aborda a detecção de objetos como um problema de regressão: ele divide a imagem em uma grade de células e, para cada célula, prevê um conjunto de caixas delimitadoras (bounding boxes) e um conjunto de pontuações de confiança (scores) para cada

objeto que ele procura. O conjunto de caixas delimitadoras e pontuações de confiança é então refinado e filtrado para gerar uma lista final de objetos detectados.

O trabalho será desenvolvido via Google Colaboratory, ou Colab, é um ambiente de desenvolvimento em nuvem gratuito baseado em Jupyter Notebook que permite que os usuários escrevam e executem códigos Python. Ele é fornecido pelo Google como um serviço de nuvem e oferece recursos para criação e treinamento de modelos de aprendizado de máquina.

2.2 DEFINIÇÃO DA BASE DE DADOS E ANÁLISE EXPLORATÓRIA

A base de dados será composta pelas três fotos abaixo, usando-as como entrada para o algoritmo detectar os gatos presentes nas imagens. A base de dados é de uso público, capturadas pelos próprios autores.

Figura 02 – Imagem I da base de dados utilizada, 1 cão e 1 gato.



Figura 03 – Imagem II da base de dados utilizada, 1 cão e 2 gatos.



Figura 04 – Imagem III da base de dados utilizada, 2 cães e 2 gatos.



Com objetivo de dificultar o processo de detecção do algoritmo, as imagens serão modificadas através do algoritmo Deep Dream, gerando padrões e texturas exageradas nas imagens como no exemplo abaixo.

Figura 05 – Exemplo da utilização do Deep Dream para modificar uma imagem.



Após alteração das imagens, serão criados 2 datasets: Dataset I, com as imagens originais; e Dataset II com as imagens alteradas. Posteriormente será realizado uma análise exploratória, os datasets serão comparados através da distribuição de seus pixels, antes e depois da aplicação do algoritmo Deep

Dream. Para isso será utilizado histograma e boxplot para analisar a frequência e a distribuição dos pixels em cada dataset.

2.3 TRATAMENTO DA BASE DE DADOS

O tratamento de base de dados de imagens é uma etapa importante no desenvolvimento de modelos de visão computacional. É essencial garantir que os dados de entrada sejam consistentes, normalizados e adequados para a tarefa de detecção ou classificação de objetos que se pretende realizar. Nesta etapa ambos os Datasets serão normalizados e dimensionados para serem usadas como entrada da Rede Neural para a detecção de objetos. Dessa forma, será utilizada a biblioteca OpenVC para converter as imagens em uma matriz numérica multidimensional.

Uma matriz numérica multidimensional é uma estrutura de dados que pode armazenar e processar dados em várias dimensões. Em visão computacional, as imagens são frequentemente representadas como matrizes numéricas multidimensionais, onde cada pixel é representado por um valor numérico. Por exemplo, uma imagem em preto e branco de 100 x 100 pixels pode ser representada como uma matriz numérica de duas dimensões de tamanho 100 x 100, onde cada elemento da matriz contém um valor numérico que representa a intensidade do pixel. Já uma imagem colorida de 100 x 100 pixels é geralmente representada como uma matriz numérica multidimensional de três dimensões de tamanho 100 x 100 x 3, onde cada elemento da matriz contém um valor numérico que representa a intensidade dos canais de cor (vermelho, verde e azul) do pixel.

2.4 MÉTODOS ANALÍTICOS UTILIZADOS

Nesta fase, teremos o momento da detecção do objeto. Para isso será utilizado a biblioteca YOLO que utiliza uma arquitetura de Rede Neural Convolucional. Durante a inferência, a rede neural divide a imagem em uma grade de células e atribui a cada célula um conjunto de caixas delimitadoras e pontuações de confiança para cada classe de objeto, permitindo a detecção em tempo real de objetos em uma imagem. Como mencionado anteriormente, outro método analítico utilizado será o Algoritmo Deep Dream.

2.3 MÉTRICA DE AVALIAÇÃO

Método utilizado para avaliar o desempenho do algoritmo de detecção será o Precision (Precisão). Neste método é observado a proporção de detecções verdadeiras positivas (TP) em relação ao total de detecções feitas (TP + falsos positivos). A precisão é uma medida da qualidade das detecções, indicando quantas das detecções feitas pelo modelo são realmente relevantes.

Para calcular a precisão (Precision) de um modelo de detecção de objetos, é necessário contar o número de detecções verdadeiras positivas (TP) e o número de detecções falsos positivas (FP). Em seguida, a precisão é calculada pela fórmula:

$$Precision = \frac{TP}{TP + FP}$$

Onde TP é o número de detecções verdadeiras positivas e FP é o número de detecções falsos positivas.

3 REPOSITÓRIO PARA DISPONIBILIZAÇÃO DOS DADOS

O armazenamento dos dados, assim como todas as informações e Notebooks Python utilizados no desenvolvimento desse trabalho, estarão disponibilizados no repositório *GitHub*:

<https://github.com/izaquenmsouza/PROJETO-APLICADO-II-MACKENZIE>

4 CRONOGRAMA

As atividades a serem cumpridas para o desenvolvimento desse trabalho são apresentadas no cronograma abaixo.

ATIVIDADES	FEV	MAR	ABR	MAI
Formação do grupo de Trabalho.	17/02			
Definição da Premissa do Projeto.	28/02			
Definição dos objetivos e Metas.	28/02			
1º Encontro Síncrono.		03/03		
Entrega da Etapa I.		13/03		
Definição da Linguagem de programação.		17/03		
Análise Exploratória dos Dados.		17/03		
Divisão da Base de Dados (Treinamento e Teste).		24/03		
Escolha e Definição das bases Teóricas dos Métodos.		24/03		
Escolha e Descrição da acurácia utilizada		24/03		

2º Encontro Síncrono.		31/03		
Entrega da Etapa II.			17/04	
Aplicação do método na base de dados			20/04	
Análise da acurácia			20/04	
Descrição e análise dos resultados preliminares			25/04	
Esboço do <i>storytelling</i>			25/04	
3º Encontro Síncrono.			28/04	
Entrega da Etapa III.				01/05
Finalização do Relatório Final				12/05
Finalização do <i>storytelling</i>				19/05
Elaboração de vídeo explicativo dos resultados finais				19/05
4º Encontro Síncrono.				26/05
Entrega da Etapa IV.				29/05

BIBLIOGRAFIA

FRANCESCHELLI, Giorgio; MUSOLESI, Mirco. Copyright in generative deep learning. **Data & Policy**, v. 4, 2022.

HUANG, Zhanchao et al. DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. **Information Sciences**, v. 522, p. 241-258, jun. 2020

JIANG, Peiyuan et al. A Review of Yolo Algorithm Developments. **Procedia Computer Science**, v. 199, p. 1066-1073, 2022

KARTHI, M. *et al.* Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset. *In*: 2021 INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING, INTELLIGENT COMMUNICATION AND SMART ELECTRICAL SYSTEMS (ICSES), 2021, Chennai, India. **2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)**. [S. l.]: IEEE, 2021

LEE, Timothy E. et al. Camera-to-Robot Pose Estimation from a Single Image. *In*: 2020 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2020, Paris, France. **2020 IEEE International Conference on Robotics and Automation (ICRA)**. [S. l.]: IEEE, 2020.

ZHOU, Tao et al. RGB-D salient object detection: A survey. **Computational Visual Media**, 7 jan. 2021