UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE RUSSAS

CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

GABRIEL BARROS ARAGÃO SILVA

USO DE *DEEP LEARNING* PARA DETECÇÃO E CLASSIFICAÇÃO DE FRUTOS DE ACEROLA EM IMAGENS DIGITAIS

RUSSAS

2022

GABRIEL BARROS ARAGÃO SILVA

USO DE *DEEP LEARNING* PARA DETECÇÃO E CLASSIFICAÇÃO DE FRUTOS DE ACEROLA EM IMAGENS DIGITAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientadora: Prof. Dr. Tatiane Fernandes Figueiredo

RUSSAS

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S58u Silva, Gabriel Barros Aragão.
Uso de deep learning para detecção e classificação de frutos de acerola em imagens digitais / Gabriel Barros Aragão Silva. – 2023.
50 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Ciência da Computação, Russas, 2023.
Orientação: Profa. Dra. Tatiane Fernandes Figueiredo.

1. Visão Computacional. 2. Redes Neurais Convolucionais. 3. Aprendizagem Profunda. I. Título.
CDD 005

GABRIEL BARROS ARAGÃO SILVA

USO DE *DEEP LEARNING* PARA DETECÇÃO E CLASSIFICAÇÃO DE FRUTOS DE ACEROLA EM IMAGENS DIGITAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Tatiane Fernandes Figueiredo (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Pablo Luiz Braga Soares
Universidade Federal do Ceará (UFC)

Prof. Dr. Marcio Costa Santos
Universidade Federal do Ceará (UFC)

RESUMO

Constantemente, a agroindústria se depara com a necessidade de intensificar e aprimorar seu processo produtivo. Nos últimos anos, no Ceará, o cenário não é diferente. A geração de grandes fazendas de cultivo de acerola no estado intensificou a busca por novas tecnologias que possam potencializar suas produções. Mais especificamente, quando se trata de acerola, a grande quantia de vitamina C presente nos frutos é um fator significativo para o crescimento da sua produção. Com esse aumento de demanda, faz-se necessário ampliar a manufatura e assim buscar melhores condições de produção dos frutos. Com o objetivo de colaborar com o estado da arte sobre aplicação de tecnologias computacionais em cultivos de acerola, este trabalho propõe a aplicação de *Deep Learning* para criação de modelos para decteção dos frutos de acerolas em imagens digitais de aceroleiras, assim como a classificação destes frutos de acordo com sua cor.

Palavras-chave: visão computacional; redes neurais convolucionais; aprendizagem profunda.

ABSTRACT

The agroindustry is constantly faced with the need to intensify and improve its production process. In recent years, in Ceará, the scenario is no different. The generation of large acerola farms in the state has intensified the search for new technologies that can enhance their production. More specifically, when it comes to acerola, the large amount of vitamin C present in the fruits is a significant factor for the growth of its production. With this increase in demand, it is necessary to expand manufacturing and thus seek better conditions for fruit production. With the aim of collaborating with the state of the art on the application of computational technologies in acerola crops, this work proposes the application of *Deep Learning* to create models for detection of acerola fruits in digital images of acerola trees, as well as the classification of these fruits according to their color.

Keywords: computer vision; convolutional neural networks; deep learning.

LISTA DE FIGURAS

Figura 1 – Sistemas de visão computacional 16
Figura 2 – Representação do *Deep Learning* (DL) na inteligência artificial 17
Figura 3 – Detecção de objetos com classes: adulto, criança e bicicleta 18
Figura 4 – Classificação de objetos com classes: adulto, criança e bicicleta 18
Figura 5 – Representação gráfica de um neurônio 19
Figura 6 – Representação gráfica das camadas de uma Redes Neurais Artificiais (RNA) 20
Figura 7 – Representação gráfica da operação das cadamas convolucionais 22
Figura 8 – Representação gráfica da operação das *pooling* 23
Figura 9 – Arquitetura de uma rede neural convolucional com um detector *Detector MultiBox Single Shot* (SSD) 26
Figura 10 – Metodologia completa aplicada nesta monografia 30
Figura 11 – Processo de aquisição de dados 31
Figura 12 – Frame com baixa capacidade de seleção 32
Figura 13 – Frame com alta capacidade de seleção 32
Figura 14 – Fotografia do processo de tratamento e análise dos dados 33
Figura 15 – Aceroleira após o processo de corte 34
Figura 16 – Aceroleira após o processo de redimensionamento 35
Figura 17 – Aceroleira após a divisão em patch 36
Figura 18 – Exemplos de patches da base de dados 37
Figura 19 – Exemplos de *patches* com marcações 38
Figura 20 – Processo de treinamento e testes do modelo 39
Figura 21 – Desenvolvimento de métricas para a etapa 1 do modelo de detecção 41
Figura 22 – Desenvolvimento de métricas para a etapa 2 do modelo de detecção 41
Figura 23 – Desenvolvimento de métricas para a etapa final de treinamento do modelo de detecção 42
Figura 24 – Desenvolvimento de métricas para a etapa 1 do modelo de classificação (*Mean Average Precisison* (mAP) = 0.702) 42
Figura 25 – Desenvolvimento de métricas para a etapa 2 do modelo de classificação (mAP = 0.7558) 43
Figura 26 – Desenvolvimento de métricas para a etapa 3 do modelo de classificação (mAP = 0.7857) 43

Figura 27 – Predições do modelo de classificação com *thresh* = 0.2 44
Figura 28 – Predições do modelo de classificação com *thresh* = 0.15 44
Figura 29 – Classificação de acerolas com modelo final (mAP = 0.7857) 45
Figura 30 – Classificação de acerolas com modelo final (mAP = 0.7857) - Parte 2 46
Figura 31 – Classificação de acerolas com modelo final (mAP = 0.7857) - Parte 3 46

LISTA DE TABELAS

Tabela 1 – Tabela de relação entre classes e objetos identificados 19
Tabela 2 – Tabela de resultados obtidos em Leite (2021) 29
Tabela 3 – Especificações de instâncias de notebooks. 39
Tabela 4 – Divisão de dados em treino e testes 39
Tabela 5 – Configuração de hiperparâmetros dos modelos 40
Tabela 6 – Comparação entre modelos de detecção 40
Tabela 7 – Comparação entre modelos de classificação 41

LISTA DE ABREVIATURAS E SIGLAS

DL	*Deep Learning*
IoU	*Intersection Over Union*
mAP	*Mean Average Precisison*
Mask-RCNN	*Mask Region-based Convolutional Neural Network*
ML	*Machine Learning*
RCNN	*Regions-based Convolutional Neural Networks*
RNA	Redes Neurais Artificiais
RNC	Redes Neurais Convolucionais
SSD	*Detector MultiBox Single Shot*
XML	*Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO 12
2 OBJETIVOS 14
2.1 Objetivo Geral 14
2.2 Objetivos Específicos 14
3 FUNDAMENTAÇÃO TEÓRICA 15
3.1 Visão Computacional e sua relação com o Aprendizado de Máquina . . 15
3.1.1 Características básicas de um sistema de visão computacional 15
3.1.2 Relacionamento com o Aprendizado de Máquina 16
3.2 Detecção e classificação de objetos em imagens digitais 17
3.3 Redes Neurais Artificiais 19
3.4 Aprendizagem Profunda e Redes Neurais Convolucionais 20
3.4.1 Camadas de Convoluções 21
3.4.2 Camadas de Pooling 22
3.4.3 Camadas Totalmente Conectadas 24
3.5 SSD: *Detector MultiBox Single Shot* 24
3.5.1 Arquitetura do SSD 25
4 TRABALHOS RELACIONADOS 27
4.1 Análise de viabilidade do uso de aprendizagem profunda para detecção de frutos de acerola em imagens RGB 27
4.2 *Deep Learning* em dois estágios para detecção e classificação de doenças em folhas de plantas com aplicação em dispositivos móveis 27
4.3 Relacionamento entre abordagens 29
5 PROCEDIMENTOS METODOLÓGICOS 30
5.1 Aquisição de dados 31
5.1.1 Gravação de vídeos das aceroleiras 31
5.1.2 Seleção de vídeos 31
5.1.3 Seleção de frames 32
5.2 Tratamento e análise dos dados 33
5.2.1 Seleção de imagens 33
5.2.2 Divisão em patches 35

5.2.3 ***Seleção de patch e marcação de acerolas*** 36
5.3 **Treinamento e testes** 38
5.3.1 ***Análise dos resultados dos testes finais realizados*** 42
6 **CONCLUSÃO** 47
6.1 **Considerações gerais** 47
6.2 **Trabalhos futuros** 48
REFERÊNCIAS 49

1 INTRODUÇÃO

A acerola é uma fruta nativa da América Central, América do Sul e das ilhas do Caribe, conhecida também como cereja das Antilhas. O Brasil é um dos poucos países que cultivam comercialmente a acerola, que foi inicialmente introduzida no estado de Pernambuco, pela Universidade Federal Rural de Pernambuco (UFRPE), em 1955 (RITZINGER; RITZINGER, 2011). Porto Rico, Havaí e Jamaica são exemplos de outros países que semeiam a fruta em escala comercial. Sendo a acerola uma fruta de grande destaque, em escala comercial, por ser uma das maiores fontes de vitamina C para o mercado farmacêutico atual. Algumas variedades chegam a ter mais de 20 vezes a quantidade do nutriente em comparação à laranja ou ao limão (SAMANTHA CERQUETANI, 2021). Além disso, a fruta vem sendo bastante utilizada para alimentação humana e até mesmo na elaboração de cosméticos (PONTES *et al.*, 2015).

Constantemente empresas de plantio de frutos de acerola se deparam com a necessidade de aumentar sua produção (GLOBO RURAL, 2014). Segundo RITZINGER e RITZINGER (2011), o aumento da demanda do produto nos mercados interno e externo vem estimulando a formação de novos plantios. Entre as diversas formas de aprimorar o procedimento produtivo dos frutos, destaca-se a atividade de mapeamento e gestão dos períodos de colheita. O conhecimento sobre a evolução das frutas e os estágios de maturação neste processo, são de grande importância para o gerencimento da recolhida dos pomos (KOIRALA *et al.*, 2019). Contudo, as aceroleiras apresentam simultaneamente frutos em diferentes estágios de formação, promovendo desuniformidade na produção, dificultando a execução dos tratos culturais, previsão e definição do ponto ideal de colheita (PONTES *et al.*, 2015).

Desta forma, um grande objetivo a ser testado e validado é o uso de técnicas de Visão Computacional para análise de imagens digitais em aceroleiras. Nesta monografia apresentamos a aplicação de *Deep Learning* para detecção e classificação de frutos de acerola. Cientificamente, os frutos de acerola são classificados em sete fases, de acordo com o peso e tamanho, desde o ponto de desenvolvimento inicial do fruto, (P1) até a maturação, (P7) (PONTES *et al.*, 2015). No entanto, devido desuniformidade presente no processo de maturação do fruto e dificuldades em captação de imagens em um ambiente não controlado real, neste trabalho consideramos apenas duas classificações relacionadas a cor do fruto, verde ou vermelha.

Os modelos apresentados nesta monografia foram desenvolvidos utilizando a arquitetura *Detector MultiBox Single Shot* (SSD), que consiste em uma arquitetura de Rede Neural Convolucional que detecta objetos em imagens usando uma única rede neural profunda. Devido

ao uso de um *framework* que não faz o uso de dados de testes para validar o treinamento dos modelos, a análise do modelo foi feita de forma manual, verificando as quantidades de predições apontadas pelo modelo final em relação a quantidade de frutos encontrados durante a obtenção dos dados para a criação do sistema de visão computacional.

Os resultados obtidos com o conjunto de teste demonstram que considerando a confiança para cada classificação dos objetos encontrados (*thresh*) próximas a 0.1, a quantidade de acerolas classificadas na cor vermelha aproxima-se consideravelmente das quantidades de frutos desta coloração encontradas na base de dados. Devido a natureza das imagens e a quantidade de acerolas verdes presentes na base de dados, a aproximação das quantidades de predições de acerolas verdes e a quantidade real de frutos encontrados, ocorreu com *thresh* de aproximadamente a 0.25.

A estrutura deste trabalho encontra-se da seguinte forma. No Capítulo 2 é apresentado o objetivo geral e os objetivos específicos desta pesquisa. O Capítulo 3 mostra os assuntos fundamentais a serem explorados neste trabalho, enquanto o Capítulo 4 apresenta os trabalhos que possuem aspectos semelhantes com esta abordagem. O Capítulo 5 descreve a metodologia e resultados obtidos durante o desenvolvimento desta pesquisa. Por fim, o Capítulo 6 apresenta as conclusões e trabalhos futuros.

2 OBJETIVOS

2.1 Objetivo Geral

Utilizar técnicas de *Deep Learning* para detectar e classificar frutos de acerolas com base na cor do fruto.

2.2 Objetivos Específicos

- Construir uma base de dados de imagens digitais de aceroleiras a partir da gravação de vídeos das plantações reais;
- Padronizar e realizar tratamentos na base de dados obtida;
- Desenvolver modelos para detecção e classificação de frutos de acerola em imagens digitais utilizando a arquitetura SSD;
- Analisar o desempenho dos modelos propostos para a detecção e classificação de frutos de acerola.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão aprentados os conceitos fundamentais a serem entendidos para a compreensão da solução que será proposta posteriormente. Nas Seções 3.1, 3.2, 3.3 serão exploradas as áreas de conhecimento fundamentais para o entendimento dessa abordagem.

3.1 Visão Computacional e sua relação com o Aprendizado de Máquina

A Visão computacional é a ciência responsável pela visão de uma máquina (MILANO; HONORATO, 2014). Trata-se de uma das linhas de pesquisa amplamente estudada nos ultimos anos, em fator da grande diversidade de procedimentos e técnicas oferecidas. (MONTANARI, 2016). Em outras palavras, essa área estuda novas formas de permitir que as máquinas possuam a capacidade de interpretar visualmente informações, ou seja, enxergar. A visão computacional possui foco na extração de informações relevantes, possibilitando reconhecimento, manipulação e analise dos objetos que compõem uma determinada imagem (BORTH *et al.*, 2014).

De acordo com Milano e Honorato (2014), cada sistema de visão computacional necessita de um conhecimento específico para resolver um determinado problema entre as diversas áreas de conhecimento. Assim, é notório que não existe uma unica forma de implementação nas aplicações que utilizam visão computacional, mas alguns procedimentos comuns podem ser adodatos em grande parte das aplicações.

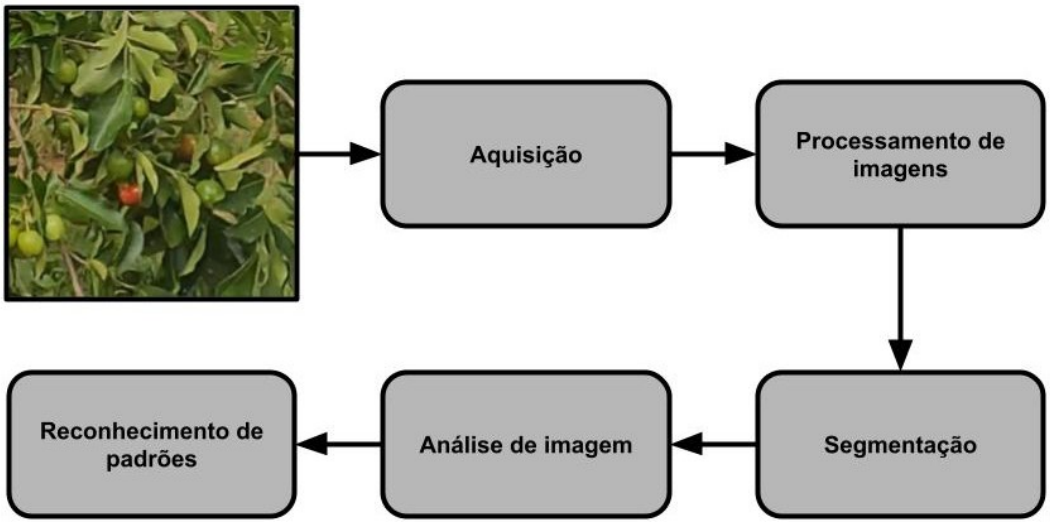
3.1.1 Características básicas de um sistema de visão computacional

De acordo com Backes e Junior (2019), um sistema de visão computacional é composto por 5 principais fases: Aquisição, Processamento de Imagens, Segmentação, Análise de Imagem e Reconhecimento de Padrões.

- Aquisição: foco na captação de imagens. Simulação da função dos olhos em comparação ao corpo humano. Nos sistemas de visão computacional, os responsáveis por esse processo são as máquinas fotográficas ou filmadoras.
- Processamento de Imagens: o objetivo deste estágio é adequar e otimizar os dados visuais adquiridos. Para a adequação podem ser usados filtros, transformações geométricas e técnicas de retirada de ruídos. Realiza-se com o propósito de oferecer imagens mais adequadas as próximas fases.

- Segmentação: divisão das imagens em regiões de interesse, ou regiões que possuem possibilidade de existência de objetos a serem detectados.
- Análise de Imagem: realiza-se a extração de características númericas das regiões de interesse que contém os objetos desejados.
- Reconhecimento de Padrões: nesta etapa os objetos de interesse são classificados e organizados em função de suas características similares, que foram obtidas na fase anterior.

Figura 1 – Sistemas de visão computacional



Fonte: Adaptado a partir de (BACKES; JUNIOR, 2019)

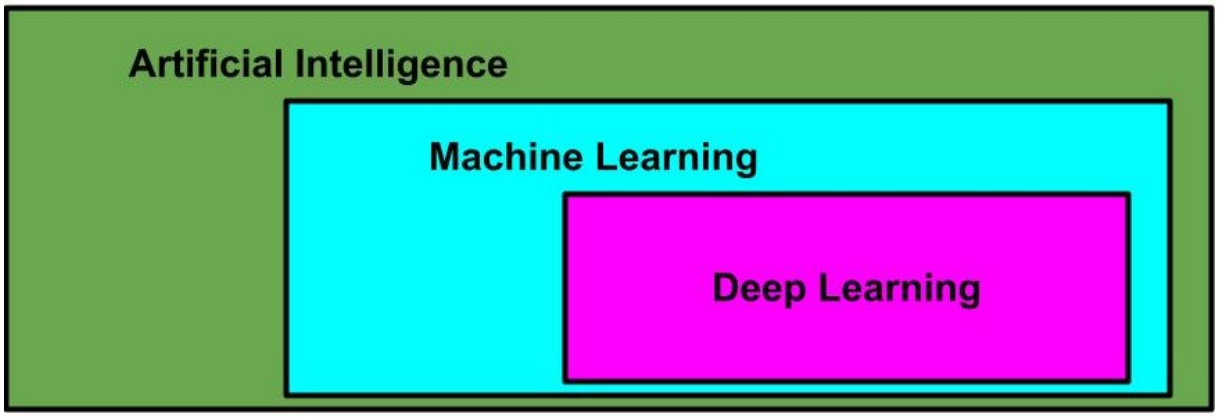
3.1.2 Relacionamento com o Aprendizado de Máquina

O relacionamento entre Visão Computacional e o Aprendizado de Máquina, do inglês *Machine Learning* (ML) pode atualmente ser visto como uma relação direta com o chamado Aprendizado Profundo, do inglês *Deep Learning* (DL). O DL é uma subárea do ML (veja a Figura 2) que se tornou mais frequentemente explorada nos últimos 10 anos, devido ao aumento da capacidade computacional e do surgimento dos grandes bancos de dados. O ML visa a construção de modelos analíticos para realizar tarefas cognitivas como detecção de objetos ou tradução de linguagem natural (JANIESCH CHRISTIAN E ZSCHECH,). No entanto, algoritmos de DL aparecem como um aprofundamento ou evolução do ML.

De acordo com LeCun *et al.* (2015), o DL permite que modelos computacionais compostos de várias camadas de processamento aprendam representações de dados com vários níveis de abstração. Assim, o acréscimo de novas camadas de processamento, possibilitaram que os algoritmos de DL aprimorassem consideravelmente o aprendizado em reconhecimento de

fala, reconhecimento visual de objetos, detecção de objetos e muitos outros domínios.

Figura 2 – Representação do DL na inteligência artificial



Fonte: Elaborado pelo autor (2022)

3.2 Detecção e classificação de objetos em imagens digitais

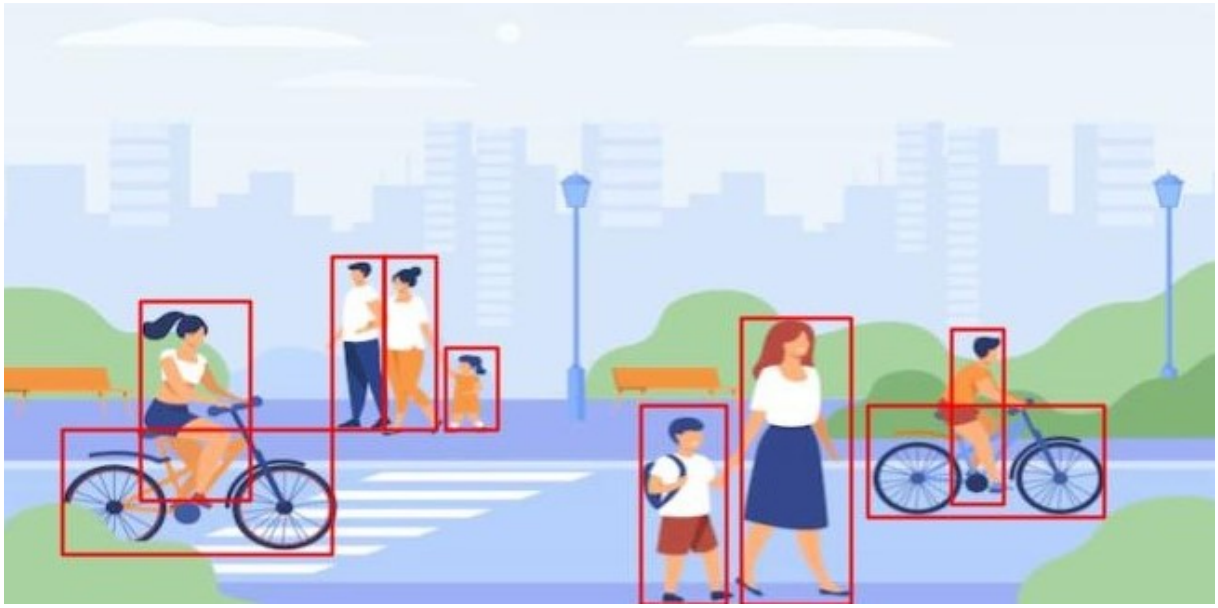
De acordo com Amit *et al.* (2020), a detecção de objetos visa determinar todas as instâncias de objetos de uma ou várias classes conhecidas. O objetivo da detecção de objetos é desenvolver modelos computacionais e técnicas que fornecem informações necessárias para a Visão Computacional (BORJI *et al.*, 2019). Ou seja, a detecção de objetos tem como principal característica identificar onde os objetos existentes em uma imagem estão, dado uma ou várias classes pré-definidas ou aprendidas de objetos que podem ser reconhecidas.

O problema de indicar a localização de um objeto (dada a classe) em uma imagem também pode ser identificado como localização de objetos. A Figura 3 representa o resultado de uma detecção de objetos em imagens digitais, considerando que o modelo de visão computacional seria desenvolvido para detectar as classes adulto, criança e bicicleta.

Por outro lado a tarefa de classificação de objetos visa prever a existência de objetos dentro das imagens. Não consiste em informar somente sim ou não para a existência de objetos em uma imagem. Além da existência dos objetos, o problema de classificação em imagens busca determinar a classe específica a qual o objeto pertence (DRUZHKOV; KUSTIKOVA, 2016).

A Figura 4 simula a execução de um algoritmo de uma classificação de objetos em imagens. Seguindo a linha de raciocínio apresentada, nesta Figura 4 pode-se observar a especificação da classe de cada objeto anteriormente detectado na Figura 3. Assim, nota-se que os algoritmos de classificação consistem em um passo posterior a detecção. Dessa maneira, o principal objetivo desse trabalho é continuar a desenvolver o estudo de técnicas de

Figura 3 – Detecção de objetos com classes: adulto, criança e bicicleta

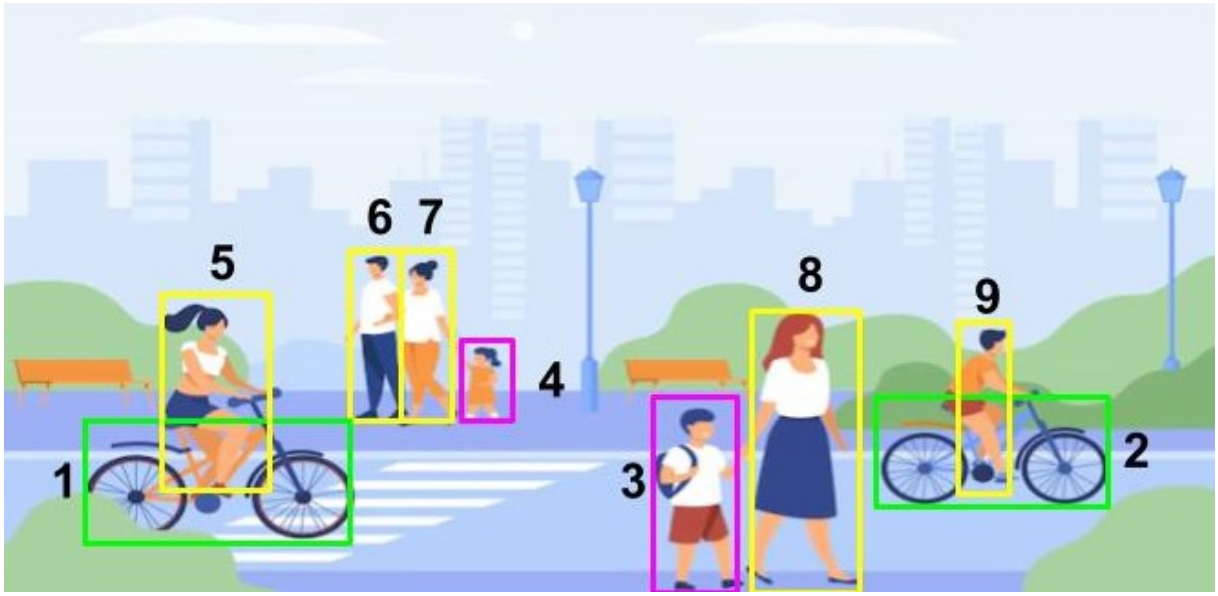


Fonte: Elaborado pelo autor (2022)

visão computacional, direcionadas ao contexto de produção e colheita de acerolas, assim como abordado em Leite (2020), no entanto com o acrescimento da classificação dos frutos de acerola por suas cores.

Na Figura 4, podem ser visto marcações em verde, amarelo e rosa, assim como para cada marcação um indentificador próximo ao retângulo colorido. Dessa maneira, a Tabela 1 traz um relacionamento entre cada objeto detectado e sua classificação.

Figura 4 – Classificação de objetos com classes: adulto, criança e bicicleta



Fonte: Elaborado pelo autor (2022).

Tabela 1 – Tabela de relação entre classes e objetos identificados

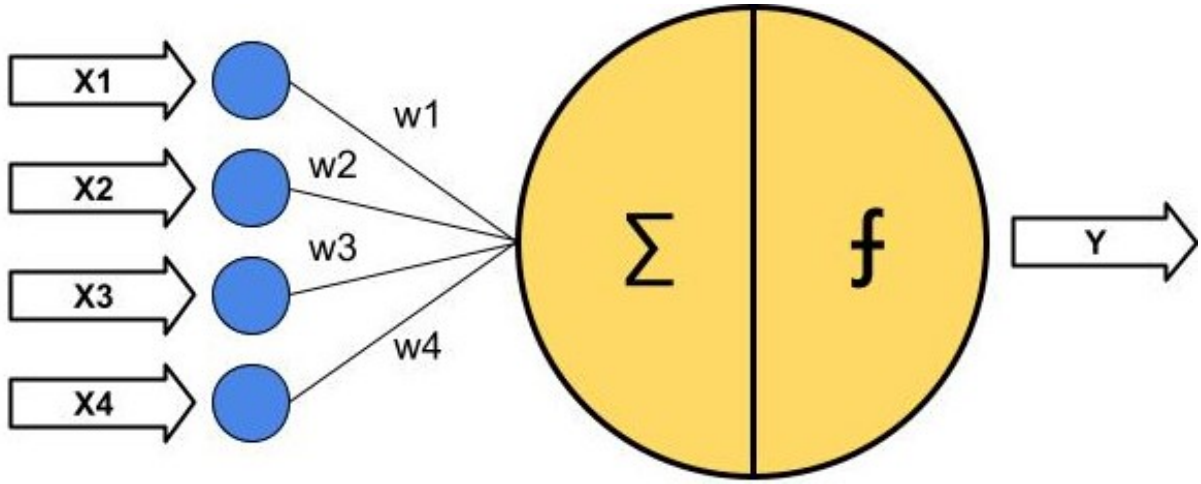
Identificador	Cor	Classe
1	VERDE	BICICLETA
2	VERDE	BICICLETA
3	ROSA	CRIANÇA
4	ROSA	CRIANÇA
5	AMARELO	ADULTO
6	AMARELO	ADULTO
7	AMARELO	ADULTO
8	AMARELO	ADULTO
9	AMARELO	ADULTO

Fonte: Elaborado pelo autor (2022).

3.3 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) vem demostrando cientificamente um grande potencial para lidar com problemas de alta complexidade, principalmente na modelagem e análise de grandes conjuntos de dados (KOVÁCS, 2002). Fitch (1944) foram os primeiros autores a apresentar um modelo matemático capaz de representar funções booleanas simulando o funcionamento dos neurônios. Seu nome e estrutura são inspirados no sistema nervoso central de um animal , imitando a maneira como os neurônios biológicos enviam sinais uns para os outros.

Figura 5 – Representação gráfica de um neurônio



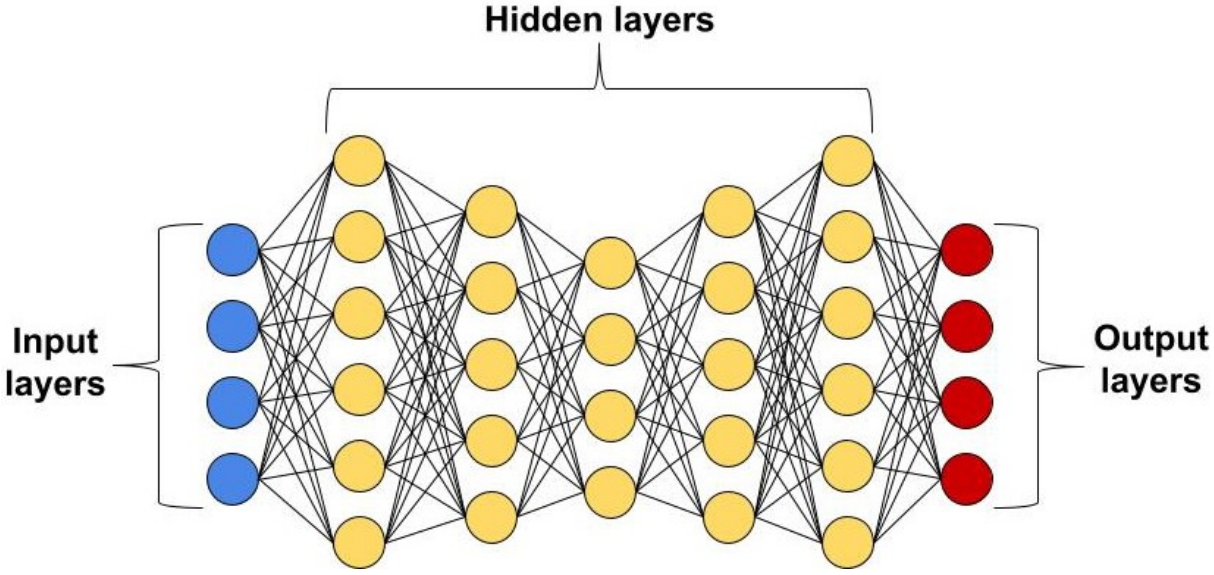
Fonte: Adaptado de (LEITE, 2020)

Desta forma, o elemento básico de uma RNA é um neurônio. Base do sistema nervoso, o neurônio é a célula com a capacidade de estabelecer conexões entre si ao receber estímulos do ambiente externo ou do próprio organismo. Nas RNA, os neurônicos são compostos por um conjunto de entradas $x = \{x1, \ldots, xn\}$, pesos $w = \{w1, \ldots, wn\}$, uma função de ativação f e uma camada de saída y. Na Figura 5, é exibida uma representação abstrata de um neurônio

com seus atributos e características citadas acima.

As RNA são compostas por camadas de nós, contendo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Os neurônios que recebem inicialmente os dados, constituem a chamada camada de entrada. Além disso, existe a camada oculta, que é representada pelos nós que recebem os dados das camadas de entrada e propagam essas informações para a frente, exceto a última camada de neurônios. A camada final, é conhecida como camada de saída, pois a partir da ativação de seus neurônios, iremos obter os resultados almejados. Na Figura 6, pode ser observada uma representação abstrata de uma RNA com 5 camadas ocultas.

Figura 6 – Representação gráfica das camadas de uma RNA



Fonte: Elaborado pelo autor (2022)

Compreender o funcionamento de uma RNA é fundamental para obter conhecimento sobre técnicas mais avançadas, como por exemplo o DL, pois elas formam uma base para todos os tipos de redes de Aprendizado Profundo.

3.4 Aprendizagem Profunda e Redes Neurais Convolucionais

Uma Rede Neural Convolucional (RNC) é um algoritmo de Aprendizado Profundo que pode captar uma imagem de entrada, atribuir importância a objetos da imagem e ser capaz de diferenciar um do outro. Basicamente, uma RNC é uma RNA mais robusta. Conforme descrito em O'Shea e Nash (2015), as Redes Neurais Convolucionais (RNC) são usadas para resolver tarefas complexas de reconhecimento de padrões em imagens digitais, o que caracteriza

a principal diferença em relação a uma RNA. A utilização de convoluções nessas redes visam tirar vantagem de como as imagens são formadas para assim tornar as redes neurais mais eficazes no processamento de imagens.

A arquitetura proposta para uma RNC possibilita a redução de parâmetros essenciais para a configuração dos modelos de aprendizagem profundo, também conhecidos como modelos de DL. Comumente a arquitetura de uma RNC é composta por camadas de neurônios classificadas em: camada de convolução, camada de *pooling* e as camadas totalmente conectadas, cada camada possui respectivamente as funções de extrair recursos, reduzir dimensões e classificar (TEUWEN; MORIAKOV, 2020). A seguir, cada tipo de camada será brevemente explorada.

3.4.1 Camadas de Convoluções

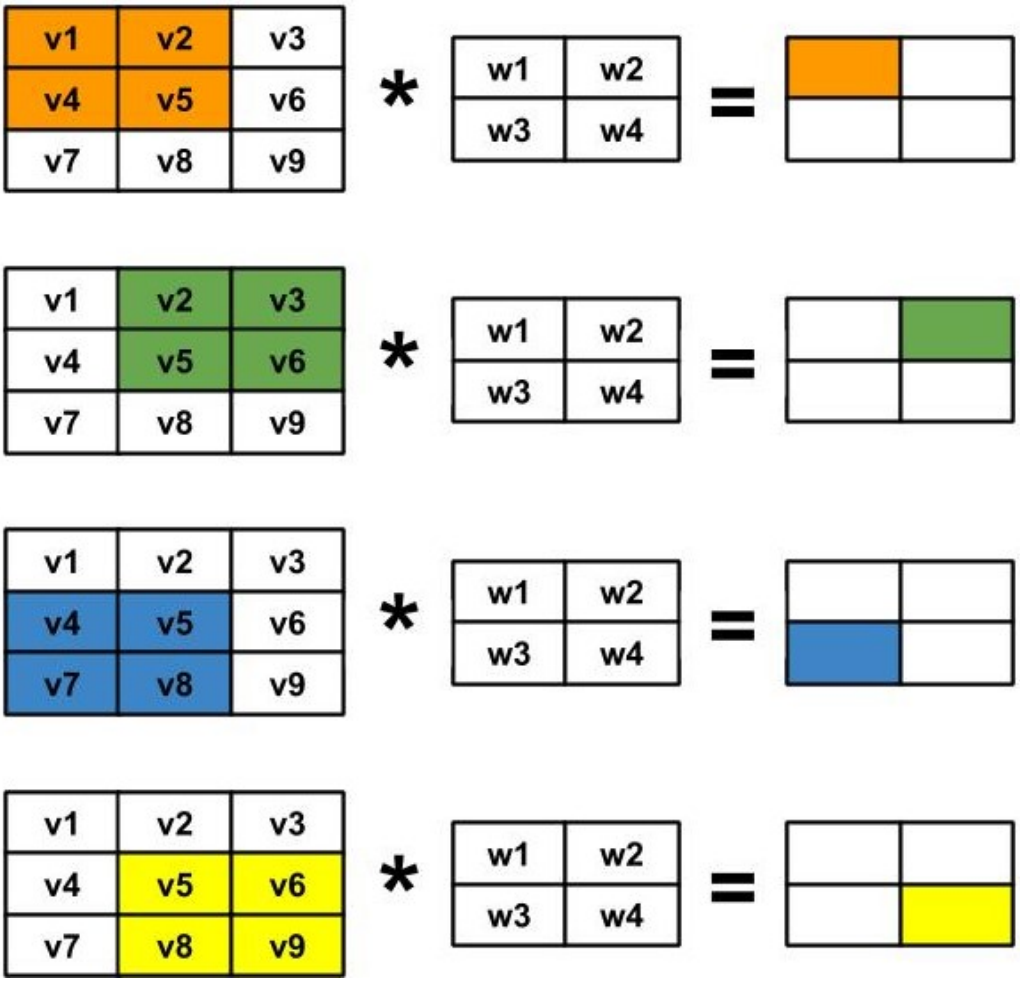
As Camadas de Convoluções tem como principal objetivo extrair recursos de alto nível dos dados de entrada e repassar para a próxima camada na forma de mapas de recursos. Uma convolução pode ser definida como um operador linear que, a partir de duas funções dadas, resulta numa terceira que mede a soma do produto dessas funções ao longo da região subentendida pela superposição delas em função do deslocamento existente entre elas.

A convolução tem o papel de fazer uma filtragem para extração de informações de interesse na imagem. Mais especificamente, o uso de filtros espaciais lineares é feito através de matrizes denominadas máscaras e a alteração dos valores presentes nesses filtros podem extrair diferentes recursos. Essa extração de recursos é feita usando operações de convolução. Durante a aplicação da convolução em uma imagem, o filtro é deslocando ao longo da imagem, como uma janela móvel, que vai multiplicando e somando os valores sobrepostos, segundo um número de passos determinados, conhecidos por *stride*.

Para faciliar o entendimento de como as camadas convolucionais trabalham, na Figura 7 será representada de forma abstrata o processo usado para calcular o mapa de recurso de uma imagem de duas dimensões (altura e largura), realizando as multiplicações e somatórios descritos antetiormente.

Considerando uma imagem de $3 \times 3 \times 1$, onde a altura e a largura têm 3 pixels, com 1 canal de cor. Um filtro de $2 \times 2 \times 1$, onde a altura e a largura têm 2 pixels, com obrigatoriamente a mesma profundidade da imagem. Deslizando o filtro sobre toda a área da imagem, considerando um *stride* igual a um, teremos uma matriz convoluída de tamanho $2 \times 2 \times 1$, pois a cada passo, é calculado o somatório do produto de todos os valores sobrepostos resultando em um único valor.

Figura 7 – Representação gráfica da operação das cadamas convolucionais



Fonte: Elaborado pelo autor (2022).

3.4.2 Camadas de Pooling

O principal objetivo das Camadas de *Pooling* é a redução das dimensões dos dados de entrada e propagação, para as camadas subsequentes, das principais características extraídas nas camadas de convoluções. Além disso, a principal vantagem de utilizá-las é a simplificação dos dados extraídos.

A redução das dimensões é feita por meio do mapaeamento de um conjunto *pixels* das imagens aplicados a alguma das possíveis operações de *pool*. É gerado um agrupamento de diversas sub-regiões das imagens, buscando destacar os principais recursos identificados dos mapas de características. Essa camada recebe cada saída do mapa de recursos da camada convolucional e prepara um mapa de características condensadas.

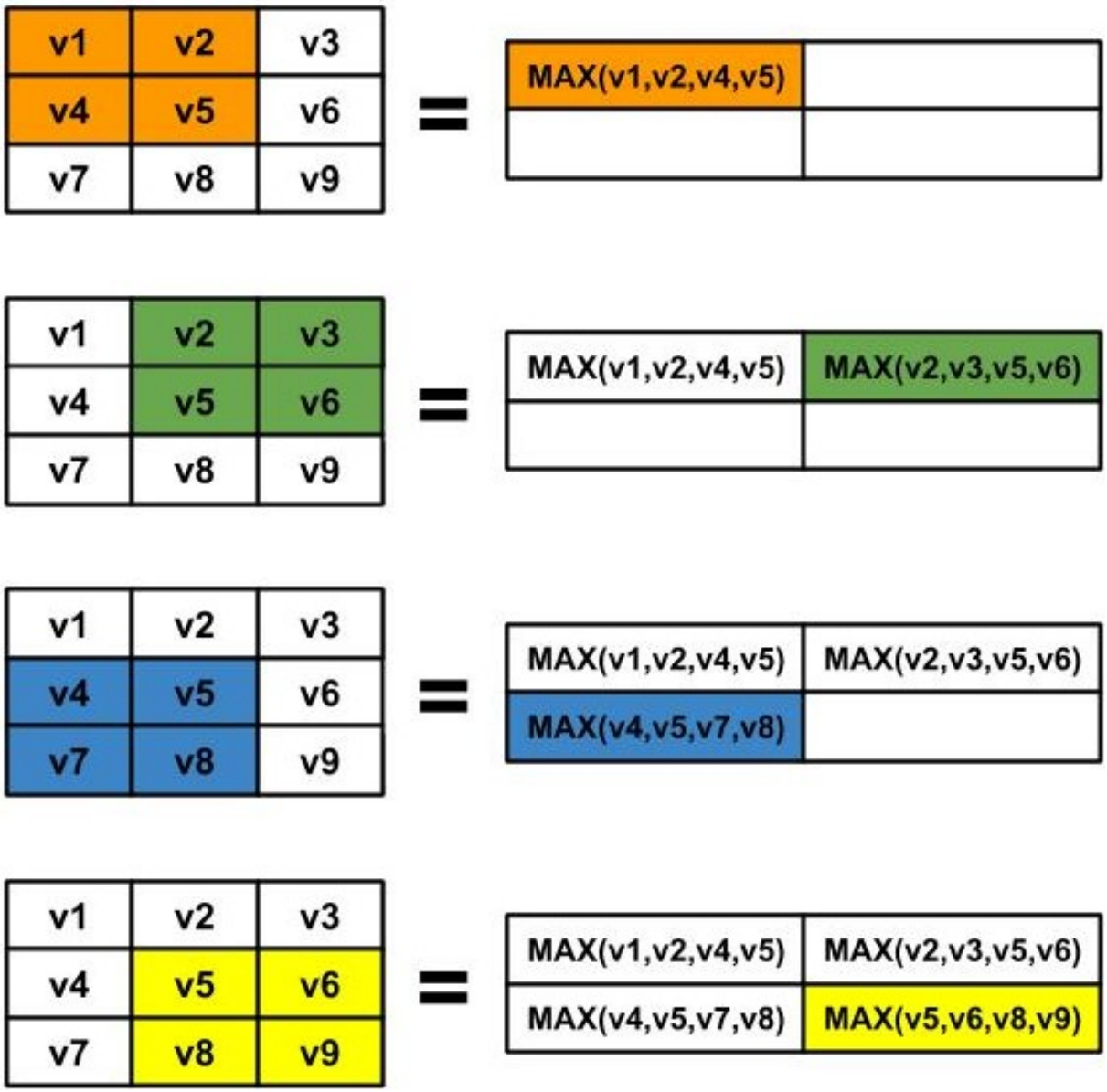
A operação de *pooling* funciona de forma similar a Camada de Convolução. É estipulado para cada camada um tamanho do *pool*, que consiste nas dimensões das sub-regiões de agrupamento. Após definido as dimensões, a lógica de agrupamento de valores segue a mesma

operação das camadas convolucionais, ou seja, aplica-se uma abordagem de janela móvel, que vai agrupando valores e calculando o mapa condensado baseado na função de *pool*.

Entre as principais funções de *pool*, destaca-se as abordagens de agrupamento médio e *Maximum Pooling*. Respectivamente, operam calculando o valor médio para cada sub-região no mapa de recursos e selecionando o valor máximo presente em cada *patch* da imagem.

Na Figura 8 será exemplificado de forma abstrata o funcionamento de uma camada de *pooling* para uma imagem dimensões $3 \times 3 \times 1$, um fator de agrupamente de 2×2 e *Maximum Pooling* como função de *pool*.

Figura 8 – Representação gráfica da operação das *pooling*



Fonte: Elaborado pelo autor (2022)

3.4.3 Camadas Totalmente Conectadas

Por fim, a arquitetura possui as camadas totalmente conectadas onde é iniciado o processo para classificar as informações extraídas pelas camadas anteriores. Nesta camada ocorre a tarefa de classificação dos objetos. As camadas totalmente conectadas são RNA que possuem todas as conexões entre os neurônios de duas camadas vizinhas na arquitetura da rede.

A camada final deste tipo de RNA comumente utiliza como método de ativação a função *softmax*. A função de ativação *softmax* é usada em redes neurais de classificação para normalizar a saída de uma rede para uma distribuição de probabilidades. Seu objetivo é fazer com que a saída de uma RNC represente a probabilidade dos dados pertencerem a uma das classes definidas.

A operação da camada totalmente conectada consiste em achatar o sub-bloco que contém os dados extraídos, ou seja, o bloco é transformado em uma única linha que contém todas as informações extraídas. Assim, mapas de características de dimensões $L \times C \times P$ (linhas, colunas e profundidade), serão tranformados em um único vetor com um total de $L \times C \times P$ posições. A Figura 6 é um exemplo de uma RNA totalmente conectada de 7 camadas, que após o processo de achatamento do mapa final de características, resultou em uma camada de entrada de 4 neurônios.

3.5 SSD: *Detector MultiBox Single Shot*

O SSD consiste em uma arquitetura de RNC que detecta objetos em imagens usando uma única rede neural profunda. Os principais sistemas de detecção de objetos por muito tempo seguiram a abordagem de criar hipóteses de caixas delimitadoras de objetos e aplicação de classificadores nos possíveis objetos detectados (LIU *et al.*, 2016). Conforme abordado em Leite (2020), as *Regions-based Convolutional Neural Networks* (RCNN) passaram por um longo processo de desenvolvimento e evolução, buscando diminuir os pontos negativos de cada abordagem proposta.

Após o processo de evolução das arquiteturas de RCNN, destaca-se a abordagem desenvolvida por Ren *et al.* (2015), a arquitetura *Faster R-CNN*. Essa perspectiva busca a diminuição do gargalo da geração de propostas de regiões de interesse presente no pensamento de RNC baseada em RCNN.

Em alternativa ao pensamento proposto nas RCNN, o SSD busca limitar um conjunto

fixo de caixas delimitadoras de diferentes tamanhos e escalas. O propósito dessa abordagem é eliminar o custo computacional com a geração de propostas de regiões, diminuir a reamostragem de recursos e a realização da detecção e classificação em uma única RNC (LIU *et al.*, 2016).

3.5.1 Arquitetura do SSD

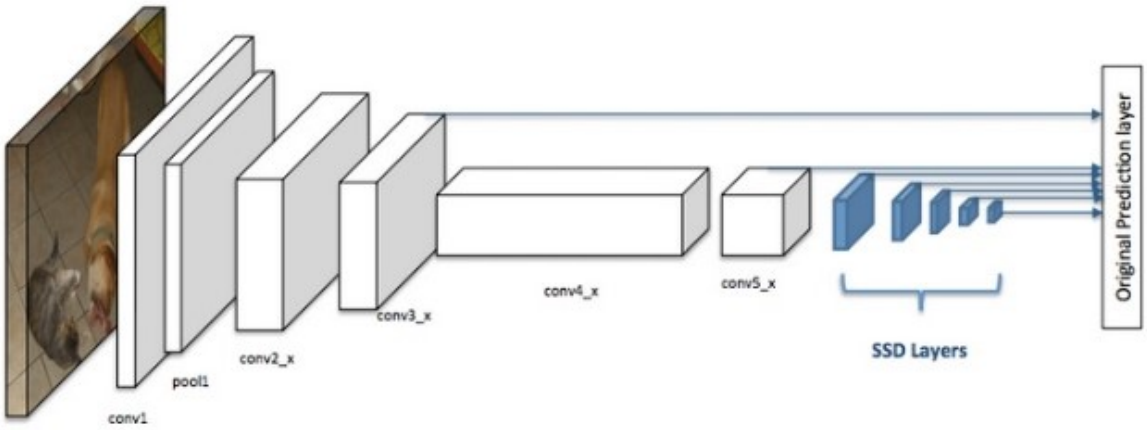
O SSD possui dois componentes principais em sua arquitetura: um modelo de *backbone* e o SSD *head*. O modelo de backbone trata-se de uma rede de classificação pré-treinada, no entanto truncada para realizar somente a tarefa de extração de recursos. Esse truncamento ocorre retirando as camadas totalmente conectadas das redes que geralmente são utilizadas para a extração de características.

Na implementação proposta em Liu *et al.* (2016), o *backbone* é composto por uma rede *ResNet* em que a camada final de classificação totalmente conectada foi removida. *ResNet* trata-se da abreviação de *Residual Networks* e consiste em uma rede neural clássica usada como base para muitas tarefas de visão computacional. O avanço fundamental decorrente do desenvolvimento das redes *ResNet* permitiu treinar redes neurais mais profundas e assim fazer um extração mais robusta de mapas de características.

A SSD *head*, trata-se de um conjunto de camadas convolucionais conectadas a este *backbone* e as saídas são interpretadas como as caixas delimitadoras. Essas camadas diminuem de tamanho progressivamente e permitem previsões de detecções em várias escalas. Cada camada do SSD *head* aplica filtros convolucionais de tamanhos diferentes gerando conjuntos diferentes de previsões e detecções.

Na Figura 9 pode ser obervada uma representação do *backbone* (caixas brancas) e o SSD *head* (caixas azuis) em conjunto, constituindo uma única RNC.

Figura 9 – Arquitetura de uma rede neural convolucional com um detector SSD



Fonte: (LIU *et al.*, 2016)

4 TRABALHOS RELACIONADOS

4.1 Análise de viabilidade do uso de aprendizagem profunda para detecção de frutos de acerola em imagens RGB

O trabalho de Leite (2020) é o ponto de partida para a criação da plataforma de tratamento inteligente adaptado ao plantio e irrigação no agronegócio. Seu principal objetivo é análisar a utilização de técnicas de processamento digital de imagens para detecção de frutos de acerola. Para isso, no trabalho citado ocorreu uma vasta exploração dos algoritmos oriundos da *Regions-based Convolutional Neural Networks* (RCNN), que consiste em uma família de modelos de aprendizado de máquina para visão computacional e especificamente detecção de objetos.

Para conduzir sua pesquisa, a metodologia proposta possui o intuito de realizar um processo de aquisição de imagens e desenvolvimento de um modelo de visão computacional, utilizando uma implementação da rede *Mask Region-based Convolutional Neural Network* (Mask-RCNN), para detectar nas imagens geradas a presença e localização de frutos de acerolas. Após a aplicação da metodologia proposta, o desempenho do modelo foi avaliado utilizado a métrica mAP, onde obteve mAP de $89,5\%$ com dados de treinamento e $92,2\%$ nos dados de teste, com o valor de $0,5$ para *Intersection Over Union* (IoU).

4.2 *Deep Learning* em dois estágios para detecção e classificação de doenças em folhas de plantas com aplicação em dispositivos móveis

O intuito da abordagem proposta por Leite (2021) é analisar o desempenho entre modelos de *Deep Learning* baseados em dois estágios em comparação aos mais tradicionas, de estágio único, no âmbito da detecção e classificação de doenças em plantas. A diferença entre a quantidade de estágios entre esses modelos decorre da separação das tarefas de detecção e classificação, que nos modelos de dois estágios, seriam realizadas por redes neurais específicas e separadas. Entre os benefícios da sepração em duas etapas destaca-se o aumento da robustez do modelo e melhor generalização durante o aprendizado. A metodologia proposta resultou no desenvolvimento de dois modelos, de um e dois estágios, como descrito acima. Cada modelo recebeu uma arquitetura própria da seguinte maneira:

- Modelo de um estágio: consistem em uma arquitetura de rede neural profunda chamada

Yolov3, utilizada no *framework* de aprendizado profundo *GluonCV*. Neste caso, a rede foi elaborada para a realização da detecção e classificação, sendo fornecidos para o treinamento as imagens e o rótulo que cada doença nas folhas.

- Modelo de dois estágios: a arquitetura proposta para este modelo é uma composição entre uma adaptação da arquitetura usada no modelo de estágio único e a arquitetura *MobileNetV2* para operação de classificação.
  - A rede destinada a tarefa de detecção foi elaborada conforme descrito no modelo de estágio único, com a adaptação de não fornercer para cada objeto um rótulo de identificação da doença. No lugar do rótulo de cada doença foi fornecido um rótulo genérico "doença", fazendo assim a transformação de uma rede de classificação, para somente detecção de objetos.
  - Para a operação de classificação foi utilizado uma segunda rede neural. A arquitetura usada para esse modelo foi a *MobileNetV2*. As caixas delimitadoras geradas como saída das rede de detecção foram extraidas das imagens e redimensionadas para 64×64. Além disso, os rótulos de cada caixa delimitadora foram fornecidos de acordo com a doença em questão, proporcionando que a rede realize a classificação.

A base de dados utilizada nesta abordagem foi obtida no site *Kaggle*. O conjunto de dados escolhido foi o *Plan Pathology* 2020, que contém um total de 1820 imagens de folhas de macieira que possuem doenças de dois tipos: ferrugem e sarna. Além disso, para a avaliação do modelo, foi utilizado um conjunto de dados auxiliar de imagens com folhas que possuem diversas doenças, no entanto, foram selecionadas somente imagens onde as folhas possuiam a doença sarna, pois este era o único tipo compatível com as doenças abordadas no trabalho de Leite (2021).

Após a realização de diversos experimentos e aplicação de várias estratégias de treinamento, observou-se que a detecção e classificação realizada em um único estágio demonstra desempenho que tende a equivalência em relação a abordagem proposta.

A natureza das imagens é algo fundamental para poder decidir uma estratégia mais eficaz para uma abordagem de visão computacional. Para modelos que precisem usar dados de diferentes fontes ou bases de dados, nas quais as imagens não foram sujeitas as mesmas condições, a abordagem de dois estágios apresentou melhor desempenho na classificação dos objetos.

Para a avaliação do desempenho dos modelos foi utilizado, assim como no presente

trabalho a mAP, que é uma métrica comumente utilizada para modelos de Visão Computacional. Os resultados foram coletados após dois ciclos de treinamento e teste dos modelos propostos no trabalho. O primeiro ciclo consistiu em realizar a execução da metodologia com imagens do mesmo conjunto de dados. No segundo ciclo, os testes foram realizados com as imagens agragadas ao conjunto de dados principal, obtidas de outra fonte como exposto anteriormente. Devido as imagens serem obtidas de diferentes fontes, estavam sujeitas a diferentes condições de qualidade. O autor destaca a importância de medir o desempenho dos modelos nessas duas condições. A seguir, pode ser observado de forma resumida os resultados obtidos no treinamento e testes dos modelos de estágio único ou duplo em cada ciclo desenvolvido na abordagem.

Tabela 2 – Tabela de resultados obtidos em Leite (2021)

Modelo	Ciclo	mAP
Um estágio	1	0.8248
Um estágio	2	0.614
Dois estágios	1	0.8172
Dois estágios	2	0.6700

Fonte: Adaptado de Leite (2021)

4.3 Relacionamento entre abordagens

Para nosso maior conhecimento, é importante ressaltar que na literatura não foram encontratos outros trabalhos utilizando visão computacinal para classificação de acerolas. Apenas o trabalho de Leite (2020), que apresenta resultados apenas para detecção de frutos de acerola em imagens RGB. Assim o espaço de trabalhos relacionados é restrito devido ao grau de inovação presente nessas abordagens.

As outra aplicações de visão computacional relacionada a detecção e classificação de características presente em frutos ou folhas no geral, são similares ao que é proposto em Leite (2021). Dessa forma, este trabalho representa um universo grande de outras propostas que fazem o uso direto de DL e ML.

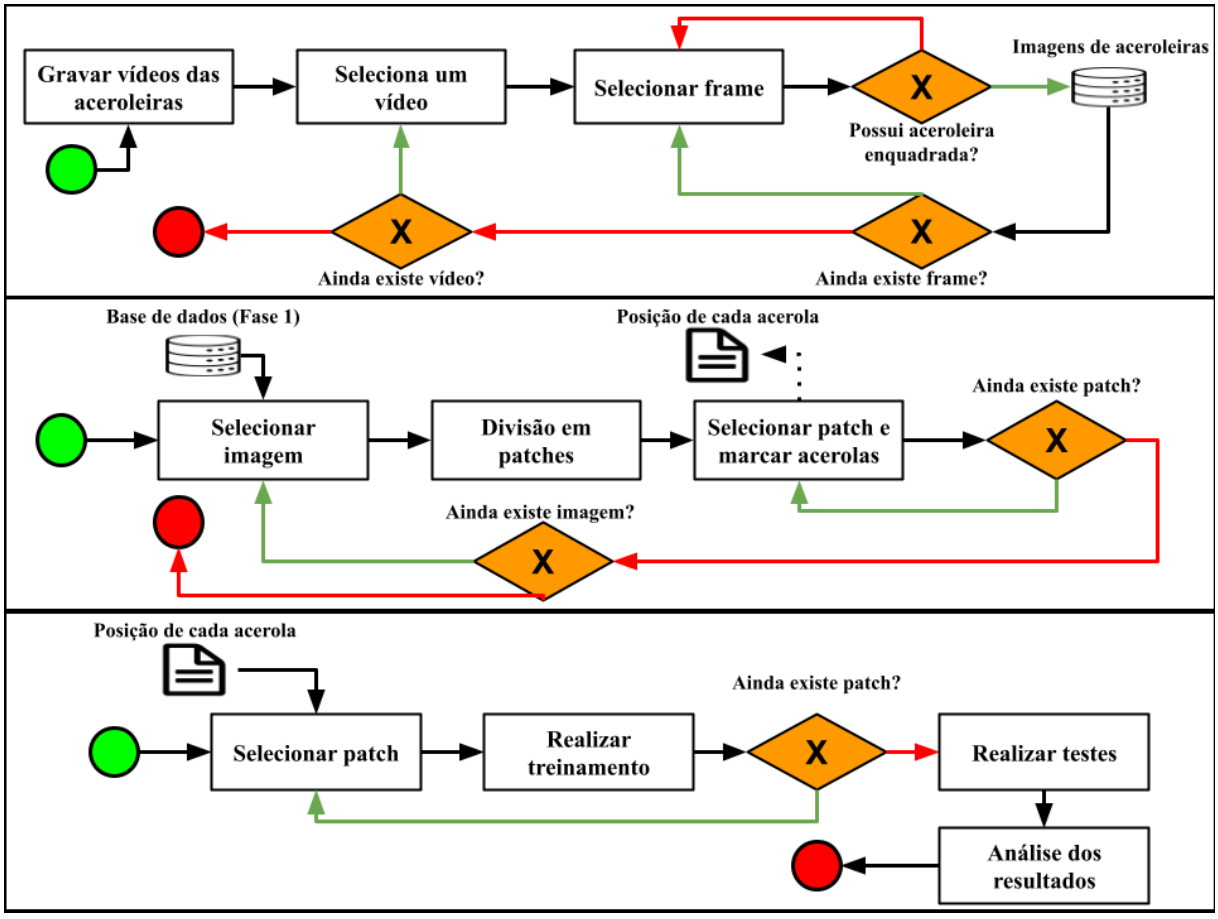
5 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo serão apresentados os passos necessários para alcançar o objetivo principal desta monografia. A metodologia está dividida em 3 processos que são representados na Figura 10. Os processos são divididos em atividades de:

1. Aquisição de dados;
2. Tratamento e análise de dados;
3. Treinamento e testes.

Os passos a serem realizados foram divididos da seguinte forma, pois consistem em processos sequenciais. Ou seja, o processo seguinte depende da finalização do anterior para consumir seus artefatos e assim gerar novos dados para serem manipulados posteriormente pelo próximo processo da cadeia. Por fim, serão realizadas atividades de análise dos resultados, com o intúito de medir a eficácia do modelo para a detecção e classificação de acerolas baseando-se na cor das frutas.

Figura 10 – Metodologia completa aplicada nesta monografia

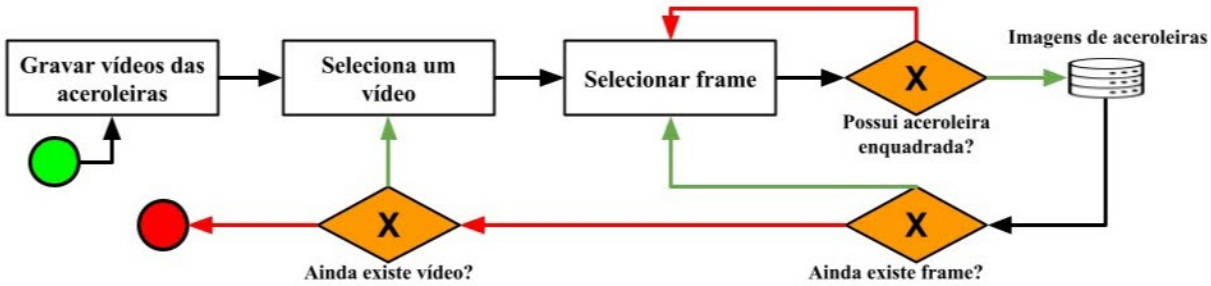


Fonte: Elaborada pelo autor (2022).

5.1 Aquisição de dados

O processo de aquisição de dados possui 3 atividades que consistem na gravação de vídeos, seleção dos vídeos e obtenção de frames especifícos com aceroleiras enquadradas. A reprentação gráfica deste passo da metodologia pode ser visto na Figura 11. Esta atividade é parte fundamental para o desenvolvimento do modelo, pois consiste no mecanismo de geração de imagens para treinamento e testes, que serão executados posteriormente.

Figura 11 – Processo de aquisição de dados



Fonte: Elaborada pelo autor (2022).

5.1.1 Gravação de vídeos das aceroleiras

A gravação de vídeos foi realizada de forma manual. Futuramente essa geração de vídeos será feita de forma automática por um veículo que possa transitar na plantação com uma câmera aclopada para realizar as gravações. Para o desenvolvimento deste trabalho, um funcionário da fazenda Mari Pobo Agropecuária utilizou uma câmera GoPro para realizar as filmagens das aceroleiras. O objetivo de filmar as árvores ao invés de simplesmente tirar fotografias é preparar a metodologia para uma futura automatização da etapa de geração de imagens. De acordo com a abordagem definida acima, foram gerados 3 vídeos gravados no campo das aceroleiras.

5.1.2 Seleção de vídeos

A atividade de seleção de vídeo, consistiu na exploração e análise dos vídeos obtidos na etapa anterior. Assim, essa etapa busca listar possíveis frames dos vídeos que podem ser canditados a se tornarem imagens da base de dados. A Figura 12 apresenta um exemplo de um possível frame que não seria selecionado para a base de dados. Já na Figura 13, apresenta um frame com grande possibilidade de seleção.

Figura 12 – Frame com baixa capacidade de seleção

Fonte: Elaborada pelo autor (2022).

Figura 13 – Frame com alta capacidade de seleção

Fonte: Elaborada pelo autor (2022).

É importante ressaltar que cada instante do vídeo pode gerar diferentes quantidades de possíveis imagens. Ainda não existem planos de automatização desta fase, pois para isso seria necessário desenvolver uma forma de detecção de árvores em vídeos, para assim selecionar automaticamente os frames ideais.

5.1.3 Seleção de frames

A atividade de seleção de frames, consistiu na filtragem das imagens geradas na etapa de seleção dos vídeos. Como apresentado anteriormente, uma aceroleira na gravação pode conter

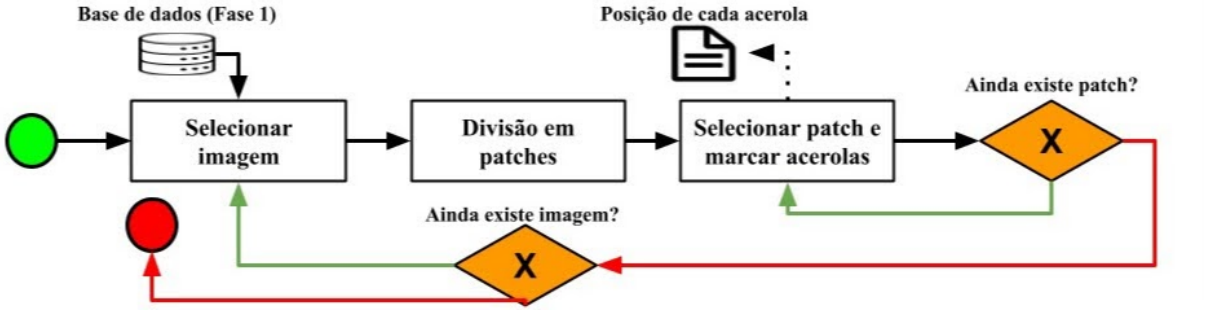
diversos frames candidados a representá-lo na base de dados. Dessa forma, buscou-se escolher o frame com melhor qualidade visual dos frutos. Essa atividade tem o intuito de selecionar as imagens mais nítidas, evitando que a base de dados seja poluída com ruídos devido ao uso de imagens que não seguem o mesmo nível de nitidez.

Após a realização da seleção do frame, finalizamos a etapa de geração de dados. Com tudo, devido a complexidade da detecção e classificação das acerolas destacada em Leite (2020), também foi realizada uma etapa de tratamento das imagens capturadas com intuito de potencializar o desempenho dos algoritmo SSD utilizado.

5.2 Tratamento e análise dos dados

A seguir na Figura 14, é apresentado o processo de tratamento dos dados realizado. As sub-atividades consistem em seleção e redimensionamento das imagens, divisão das imagens em *patches* e marcação do posicionamento dos frutos em cada *patch* selecionado.

Figura 14 – Fotografia do processo de tratamento e análise dos dados



Fonte: Elaborada pelo autor (2022).

5.2.1 Seleção de imagens

A etapa de seleção de imagens teve como principal objetivo padronizar o tamanho das imagens geradas. Dessa forma, o processo de padronização consiste redimensionar todas a imagens para as dimensões de 3584×2048. Este dimesionamento foi definido buscando preservar a qualidade e o detalhamento presente nas imagens, além de garantir um tamanho razoável que viabilizá-se a estratégia de divisão em *patchs* realizada posteriormente.

Algumas imagens da base de dados necessitaram ser recortadas para ajustar o enquadramento da árvore, buscando eliminar árvores que não estão no primeiro plano ou espaços vazios nas laterais da planta. A seguir, na Figura 15 é possível observar uma aceroleira que

necessitou ser recortada para ajustar o enquadramento da árvore. Na Figura 16, pode-se observar a mesma aceroleira, porém redimensionada para as medidas citadas anteriormente.

Figura 15 – Aceroleira após o processo de corte

Fonte: Elaborada pelo autor (2022).

Figura 16 – Aceroleira após o processo de redimensionamento

Fonte: Elaborada pelo autor (2022).

5.2.2 Divisão em patches

A atividade de divisão em *patch* é amplamente utilizada em diversas aplicações de Visão Computacional (BARGOTI; UNDERWOOD, 2017; KOVALEV *et al.*, 2016). Um dos principais benefícios desta técnica é viabilizar a detecção visual de objetos pequenos em imagens de grande proporção em relação aos alvos. Além disso, o tamanho das entrada nos algoritmos de Visão Computacional, é diretamente proporcional ao custo operacional das atividades de detecção e classificação.

A técnica de divisão em *patch* consiste em subdividir uma imagem em imagens menores. A utilização desse método em Visão Computacional nos permite realizar os treinamentos em imagens menores, ou seja, em pedaços de mesmo tamanho de uma imagem original. Para exemplificar, para a execução do algoritmo SSD sem a divisão em patchs, visto que as imagens foram padronizadas nos tamanhos de 3584×2048, seria necessário alocação de matrizes de $3584 \times 2048 \times 3 = 22.020.096$ posições o que acarretaria em um grande gasto de memória e processamento.

Dessa forma, as imagens da base de dados foram dividas no formato *grid*. O termo *grid* refere-se a um elemento técnico que é constituído por linhas verticais e horizontais ou quadrados e retângulos. O *grid* tem como principal objetivo auxiliar na ordenação, distribuição, alinhamento e dimensão de imagens, textos, formas e outros elementos.

Portanto, as imagens foram divididas em *patches* de tamanho 512×512, seguindo a abordagem utilizada inicialmente por Leite (2020). A divisão resultou em um *grid* de 4 linhas (2048/512) e 7 colunas (3584/512) para cada imagem, como pode ser observado na Figura 17

Figura 17 – Aceroleira após a divisão em patch

Fonte: Elaborada pelo autor (2022).

5.2.3 Seleção de patch e marcação de acerolas

O processo de seleção de *patches* e marcação dos frutos tem como objetivo a geração da entrada do algoritmo SSD utilizado neste trabalho. Além da base de dados, é necessário informar ao algoritmo a localização nas imagens dos objetos a serem detectados, denominada na literatura como marcação, para que assim o modelo possa aprender a detectar e classificar de acordo com essas informações.

A marcação das coordenadas de cada acerola foi feita utilizando a ferramenta *open-source Pychet Labeller*, desenvolvida em *Python* e diponível no repositório *https://github.com/acfr/pychetlabeller*. Todos os frutos encontrados nas imagens receberam uma caixa delimitadora de 4 coordenadas denominadas $Xmin, Ymin, Xmax, Ymax$ tendo como resultado arquivos no formato *Extensible Markup Language* (XML), além disso foi atribuida a cada marcação um *label* de "acerola", independendo de sua cor. A execução deste processo gerou os seguinte insumos

para o presente trabalho:

1. 100 imagens de aceroleiras enquadradas;
2. 2800 *patches*;
3. 846 arquivos XML;
4. 1756 regiões de interesse com a *label* "acerola".

Em um segundo momento, todas as regiões de interesse que foram identificadas como “acerola”, foram segregadas em dois conjuntos de acordo com a cor do fruto, gerando assim uma novo insumo para o desenvolvimento do modelo de classificação proposto. Dessa maneira, foram geradas para a base de dados, 1388 regiões de interesse com a *label* “acerola_verde” e 372 denominadas “acerola_vermelha”.

Na Figura 18 podem ser visto exemplos de alguns *patches* utilizados no processo de marcação. Já na Figura 19, pode-se observar a imagens com a marcação de cada fruto.

Figura 18 – Exemplos de patches da base de dados

Fonte: Elaborada pelo autor (2022).

Figura 19 – Exemplos de *patches* com marcações

Fonte: Elaborada pelo autor (2022).

5.3 Treinamento e testes

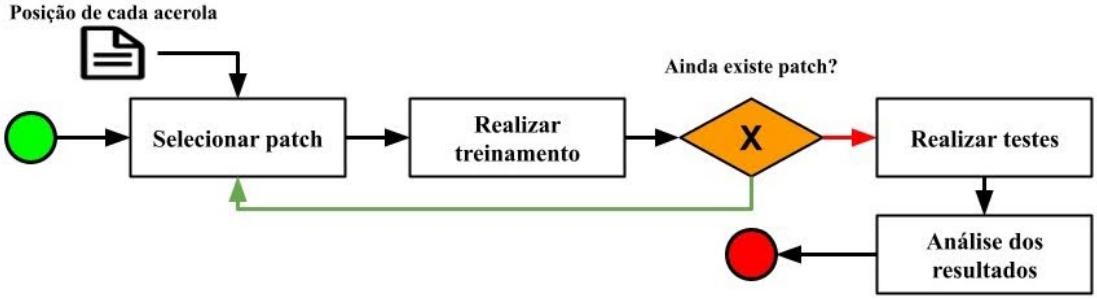
O processo de treinamento e testes teve como objetivo executar o algoritmo SSD. Para a criação dos modelos foi utilizada a plataforma de computação em nuvem, AWS (*Amazon Web Services*), através do *framework Amazon SageMaker*. Assim, nesta etapa o foco do trabalho foi executar os passos básicos de treinamento de uma modelo de Aprendizado de Máquina Supervisionado comumente utilizados na literatura:

1. Alimentação do modelo com dados de treinamento e validação;
2. Etapa de aprendizagem através da associação das imagens às classes pré-estabelecidas;
3. Realização de previsões utilizando o conjunto de teste.

A seguir, na Figura 20, podemos observar uma representação deste processo. As etapas internas de execução do algoritmo SSD aplicado ocorreram conforme explicitado na Capítulo 3, que trata da abordagem proposta em Kelner *et al.* (2013).

Para o processo de treinamento do modelo foram usadas três instâncias de máquinas virtuais. O algoritmo SSD foi codificado na linguagem de programação *Python*. As configurações

Figura 20 – Processo de treinamento e testes do modelo



Fonte: Elaborada pelo autor (2022).

técnicas de cada máquina são representados na Tabela 3.

Tabela 3 – Especificações de instâncias de notebooks.

Tipo	Processador	RAM	GPU
ml.t2.medium	Intel(R) Xeon(R) E5-2676 v3 @ 2.4 GHz	4 GB	X
ml.p3.2xlarge	Intel(R) Xeon(R) E5-2686 v4 @ 2.3 GHz	61 GiB	16 GiB
ml.m4.xlarge	Intel(R) Xeon(R) E5-2676 v3 @ 2.4 GHz	16 GiB	X

Fonte: Elaborado pelo autor (2022).

Com a utilização dessas máquinas virtuais não foram encontradas limitações relacionadas ao tempo de acesso e tempo limite de inatividade durante os treinamentos, algo que foi destacado como fatores dificultadores em Leite (2020), quando foi utilizado outra plataforma de computação em nuvem.

Para cada modelo, as imagens de treino e testes eram segregadas de forma aleatória. Durante essa divisão, optou-se por aproximar as quantidade de 75% dos dados para o conjunto de treino e o restante para os dados de teste, o número de imagens para cada etapa é apresentado na Tabela 4.

Tabela 4 – Divisão de dados em treino e testes

Conjunto de dados	Modelo de detecção	Modelo de classificação
Treinamento	637	653
Testes	209	195

Fonte: Elaborado pelo autor (2022).

Para cada abordagem desenvolvida foi utilizado o mesmo padrão de configuração de hiperparâmetros para a execução dos treinamentos, com exceção das métricas que obrigatoriamente deveriam ser diferente devido a natureza de cada modelo. A configuração dos hiperparâmetro pode ser observada na Tabela 5.

O processo de treinamento do modelo de detecção ocorreu em 3 diferentes etapas.

Tabela 5 – Configuração de hiperparâmetros dos modelos

Hiperparâmetro	Modelo de detecção	Modelo de classificação
num_classes	1	2
num_training_samples	643	653
mini_batch_size	8	8
epochs	32/64/64	16/32/64
base_network	resnet-50	resnet-50
learning_rate	0.001	0.001
optimizer	sgd	sgd
momentum	0.9	0.9
weight_decay	0.0005	0.0005
image_shape	512	512

Fonte: Elaborado pelo autor (2022).

Com o intuito de obter melhores resultados, foi utilizada uma conduta de compartilhamento de conhecimento entre cada treinamento. Dessa maneira, ao iniciar um novo treinamento, os resultados obtidos no processo anterior serviam como base para a próxima etapa.

Na Tabela 6 podem ser observados e comparados os resultados de cada etapa do treinamento, assim como algumas métricas amplamente utilizadas para análise do desempenho de modelos de Aprendizado de Máquina. Além disso, na Figura 21, Figura 22 e Figura 23, pode-se observar o desenvolvimento das métricas disponibilizadas no *Amazon SageMaker*, em relação às épocas de treino para cada etapa de treinamento do modelo de detecção de acerolas.

Tabela 6 – Comparação entre modelos de detecção

Treinamento	Épocas	mAP inicial	Melhor mAP	Tempo (sec)	*cross_entropy*	*smooth_l1*
Etapa 1	32	X	0.7357	946	0.5255	0.2709
Etapa 2	64	0.7357	0.7855	1588	0.4772	0.1796
Etapa 3	64	0.7855	0.8103	1573	0.4413	0.1463

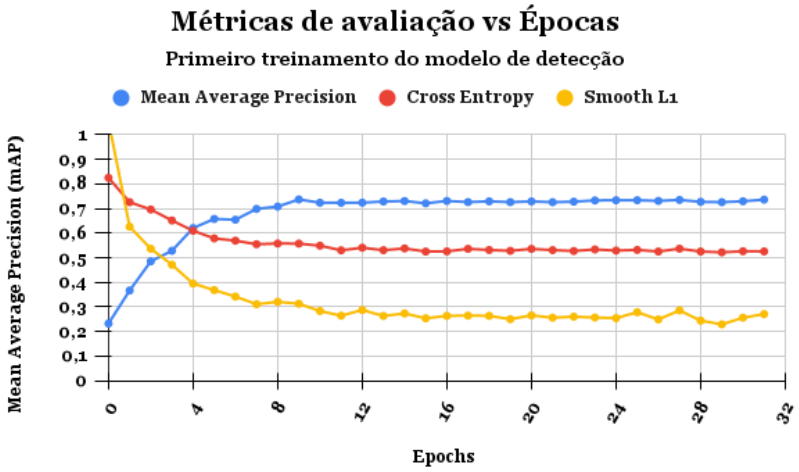
Fonte: Elaborado pelo autor (2022).

No processo de criação do modelo de classificação de frutos de acerola também foi utilizado a técnica de compartilhamento de conhecimento entre cada modelo. Optou-se por seguir as mesmas configurações de hiperparâmetros usadas nos procedimentos anteriores, com exceção do número de classes a serem detectadas no modelo, quantidade de imagens de treinamento e configuração da quantidade de epócas para cada etapa de aprendizagem do modelo.

Assim, de forma similar, o modelo detecção o modelo de classificação também foi treinado em 3 etapas. Cada etapa da treinamento foi executada por 16, 32 e 64 épocas respectivamente, com o intuito de uma maior exploração dos dados de treino a medida que a precisão do modelo demora mais para evoluir.

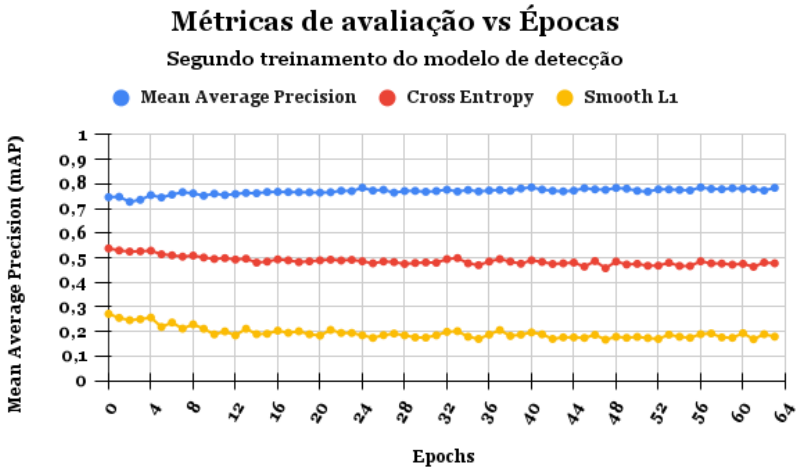
Após todos os treinamento e coleta de métricas disponibilizadas pelo *SageMaker*,

Figura 21 – Desenvolvimento de métricas para a etapa 1 do modelo de detecção



Fonte: Elaborada pelo autor (2022).

Figura 22 – Desenvolvimento de métricas para a etapa 2 do modelo de detecção



Fonte: Elaborada pelo autor (2022).

é possível acompanhar de forma matemática o crescimento do conhecimento do modelo de classificação na Tabela 7. Assim como na metodologia usada no modelo de detecção, os conjuntos de dados de treino e testes foram mantidos durante todas as etapas, evitando assim variação de aprendizado pela diferenciação dos dados.

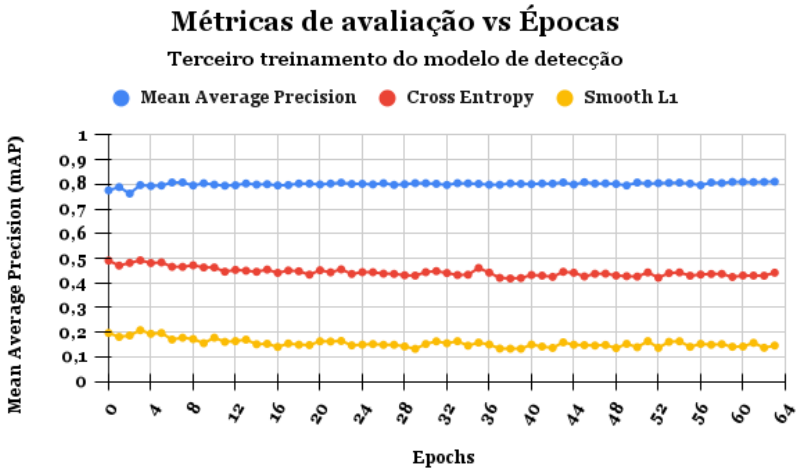
Tabela 7 – Comparação entre modelos de classificação

Treinamento	Épocas	mAP inicial	Melhor mAP	Tempo (sec)	*cross_entropy*	*smooth_l1*
Etapa 1	16	X	0.702	691	0.5933	0.256
Etapa 2	32	0.702	0.7558	1005	0.538	0.187
Etapa 3	64	0.7558	0.7857	1648	0.4668	0.159

Fonte: Elaborado pelo autor (2022).

Na Figura 24, Figura 25 e Figura 26, pode-se observar a evolução das métricas

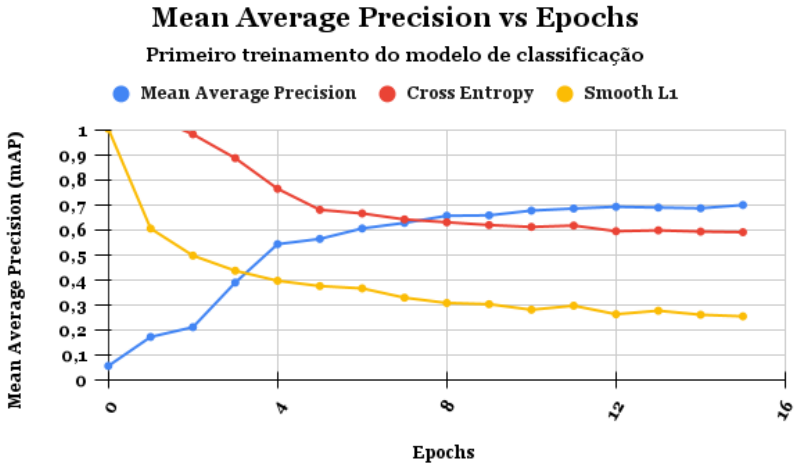
Figura 23 – Desenvolvimento de métricas para a etapa final de treinamento do modelo de detecção



Fonte: Elaborada pelo autor (2022).

disponibilizadas no *Amazon SageMaker* em relação às épocas de treinamento de cada modelo de classificação, repectivamente.

Figura 24 – Desenvolvimento de métricas para a etapa 1 do modelo de classificação (mAP = 0.702)

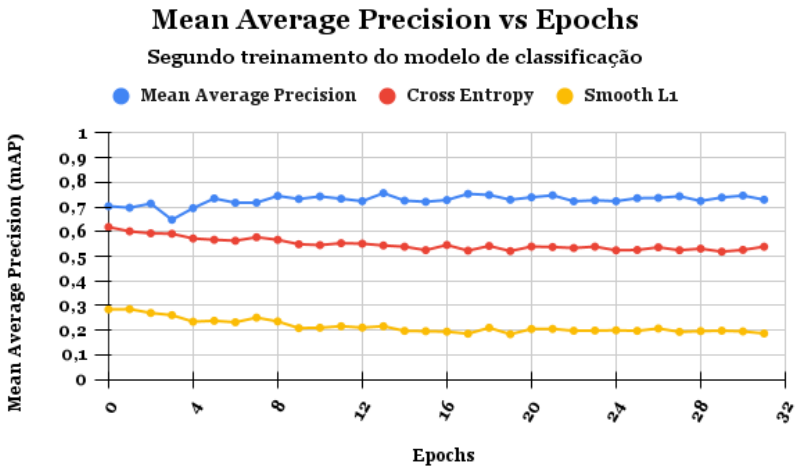


Fonte: Elaborada pelo autor (2022).

5.3.1 Análise dos resultados dos testes finais realizados

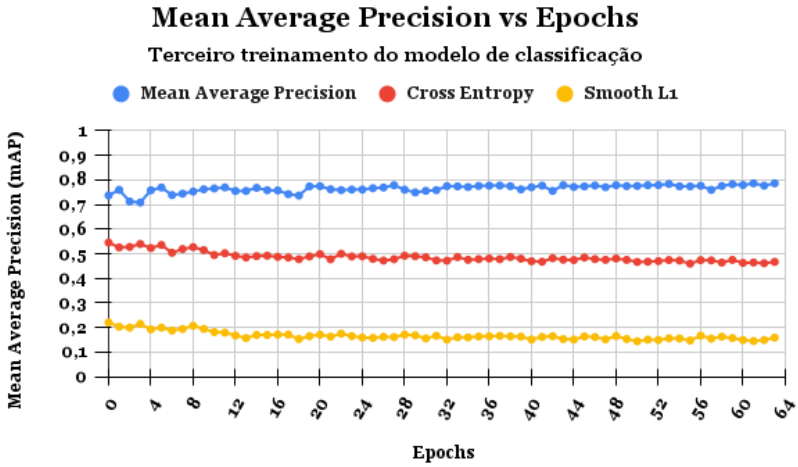
Os modelos gerados após a etapa de treinamento foram armazenadas automaticamente na plataforma AWS, através do serviço de banco de dados *Amazon S3*, sendo possível submeter as imagens de aceroleiras do conjunto de teste, para realização do processo de detecção e classificação de frutos de acerola.

Figura 25 – Desenvolvimento de métricas para a etapa 2 do modelo de classificação (mAP = 0.7558)



Fonte: Elaborada pelo autor (2022).

Figura 26 – Desenvolvimento de métricas para a etapa 3 do modelo de classificação (mAP = 0.7857)

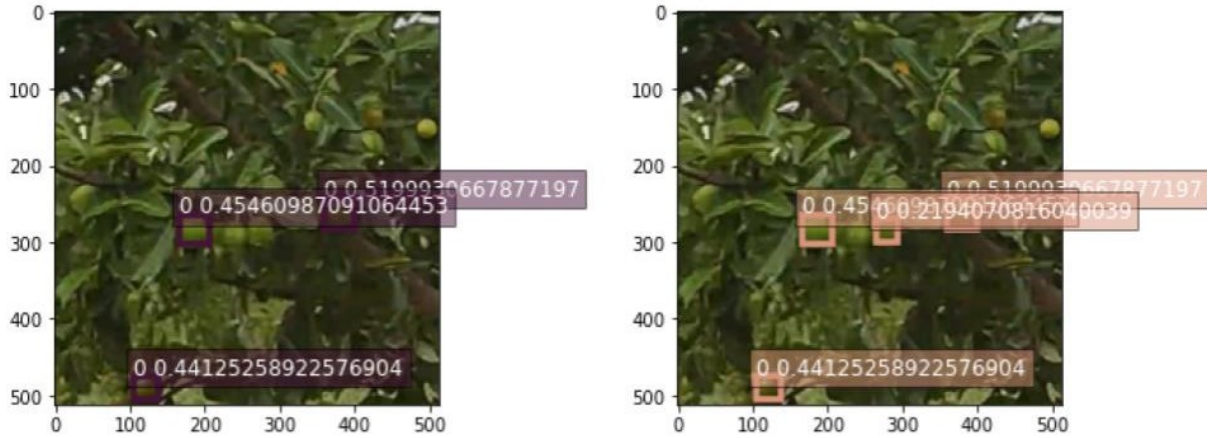


Fonte: Elaborada pelo autor (2022).

Desta forma, podemos visualizar e analisar o funcionamento dos modelos e gerar exemplos visuais das predições realizadas. A Figura 27 e a Figura 28, apresentam as classificações realizadas. Neste caso, cada acerola detectada recebeu o valor de uma *label*, 0 para verde e 1 para vermelha, além de um valor decimal que representa o *thresh* da detecção. O valor de *thresh* representa a confiança para cada classificação de que cada objeto correspondente a classe a qual a ele foi atribuida no processo de predição.

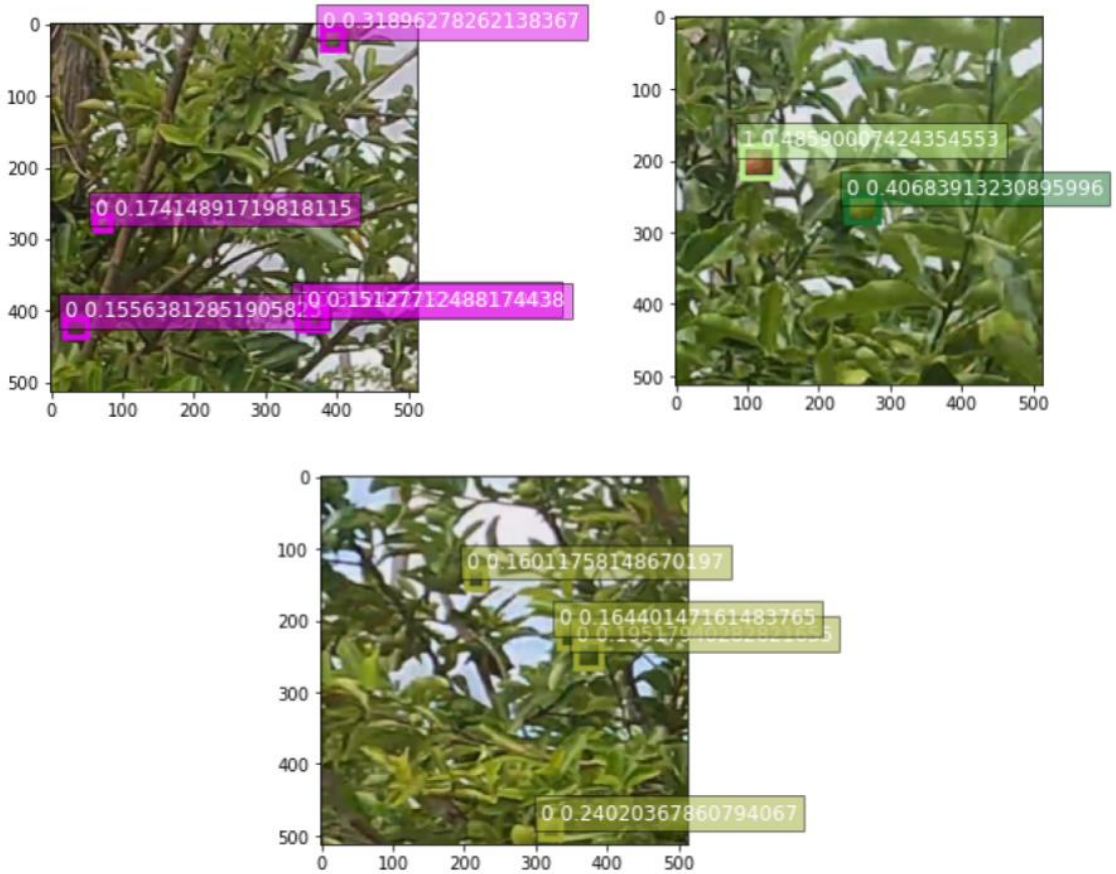
Foi realizada um teste com 40 imagens do conjunto de testes para análise de acertos e erros do modelo de classificação, onde o modelo detectou e classificou um total de 50 acerolas vermelhas e 38 frutos verdes. Considerando *thresh* próximas a 0.1, a quantidade de acerolas

Figura 27 – Predições do modelo de classificação com *thresh* = 0.2



Fonte: Elaborada pelo autor (2022).

Figura 28 – Predições do modelo de classificação com *thresh* = 0.15

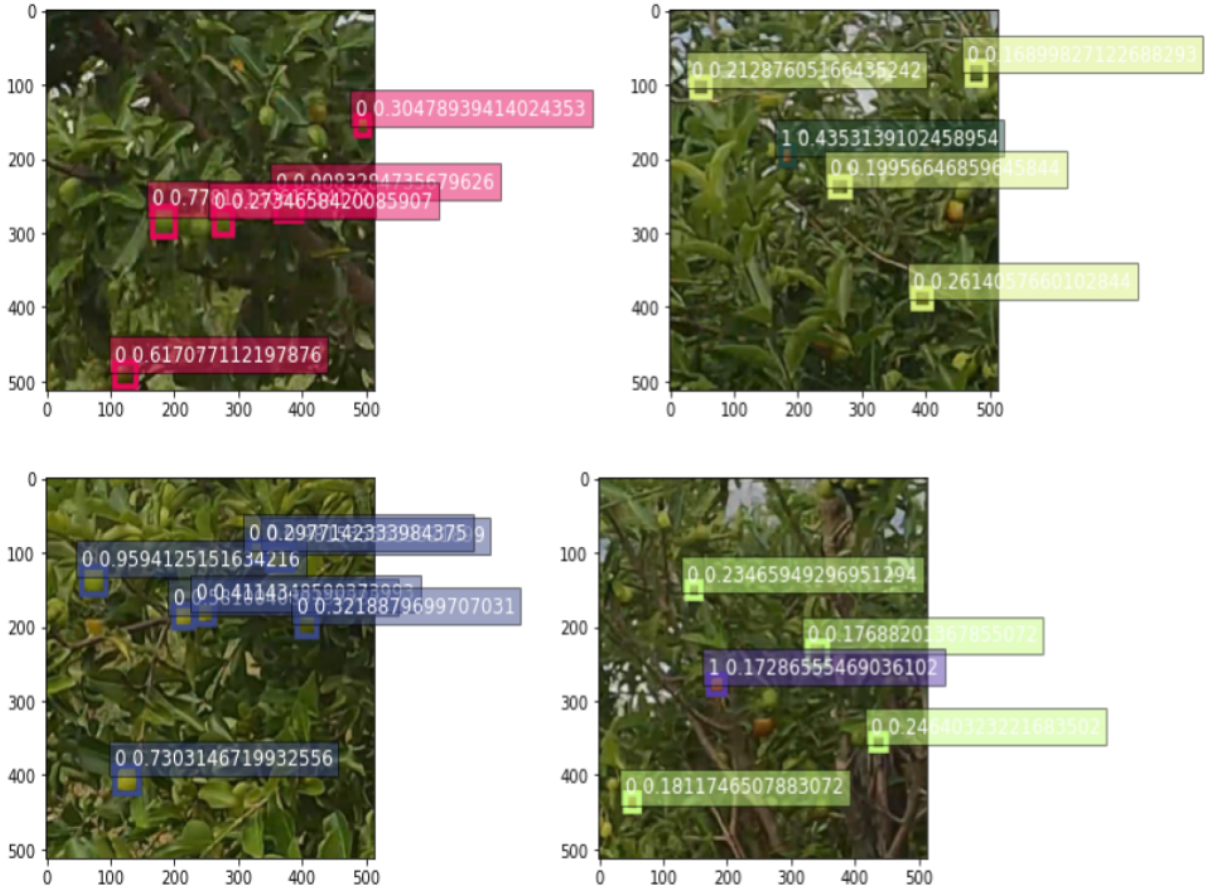


Fonte: Elaborada pelo autor (2022).

classificadas na cor vermelha aproxima-se consideravelmente das quantidades de frutos desta coloração encontradas na base de dados, que eram de 52. Em relação as classificações de frutos verdes, com *thresh* de aproximadamente 0.25, o modelo obteve 32 predições de acerolas verdes, enquanto a quantidade real na base de dados era de 38 frutos verdes. A seguir a Figura 29, a Figura 30, a Figura 31 apresenta algumas imagens do conjunto de teste que passaram pelo

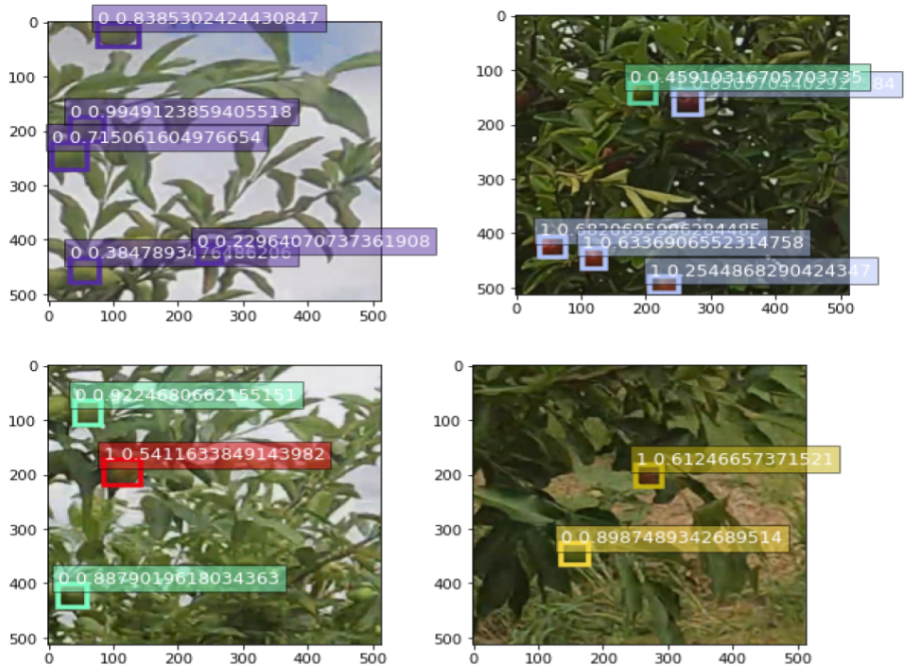
processo de classificação utilizando o modelo de melhor mAP.

Figura 29 – Classificação de acerolas com modelo final (mAP = 0.7857)



Fonte: Elaborada pelo autor (2022).

Figura 30 – Classificação de acerolas com modelo final (mAP = 0.7857) - Parte 2



Fonte: Elaborada pelo autor (2022).

Figura 31 – Classificação de acerolas com modelo final (mAP = 0.7857) - Parte 3

Fonte: Elaborada pelo autor (2022).

6 CONCLUSÃO

Neste capítulo serão descritas as considerações e lições aprendidas a respeito da metodologia desenvolvida na atual abordagem, para classificação de frutos de acerola de acordo com a cor dos frutos, assim como os resultados obtidos durante a execução da mesma.

6.1 Considerações gerais

Neste trabalho foi proposta uma metodologia para classificação de frutos de acerola em imagens digitais, de acordo com suas cores, verde ou vermelha. Para tal, utilizou-se a arquitetura SDD para geração dos modelos. Em comparação aos desafios encontrados em outras abordagens que utilizam Visão Computacional para detecção de acerolas em imagens RGB, pode-se afirmar que este trabalho consiste em uma evolução do que foi desenvolvido em Leite (2020).

Como resultado principal o modelo de classificação apresentou um taxa de acerto de 78.5% para classificação de acerolas (verde ou vermelha) no conjunto de treinamento. Para o conjunto de testes, considerando diferentes valores de confiança das predições, foi possível aproximar as quantidades de frutos classificados, mas ainda é necessário descobrir como realizar a avaliação do modelo com toda a base de dados, usando o *Amazon SageMaker*, para obter um *mAP* final para o conjunto de testes, visto que o próprio *framework* não realiza essa atividade de forma explícita aos desenvolvedores.

Portanto pode-se considerar que a metodologia proposta contribuiu com o estado da arte de forma significativa, atuando sobre o processo de criação de uma nova base de dados normalizada, realizando também a exploração do *framework Amazon SageMaker* para resolução do problemática descrita nesta monografia. Para tal contribuição, como descrito em 5.1, realizou-se a obtenção de dados de uma única fonte, os campos de aceroleiras da fazenda Mari Pobo Agropecuária, onde todas as imagens geradas estariam sujeitas às mesmas condições de ambiente, reduzindo assim a variabilidade de distanciamento do fruto, condições de iluminação, dentre outros aspectos. Além disso, como proposto 5.3, a exploração do *Amazon SageMaker* apresentou grande potencial para resolução de problemas similares ao proposto neste trabalho.

Portanto, pode-se concluir que as principais dificuldades encontradas consistiram na realização do processo de marcação das regiões de interesse para cada modelo desenvolvido, visto que nesta abordagem esse processo foi feito duas vezes e em grande parte de pesquisas

relacionadas a detecção e classificação está etapa é inevitável. Além disso, ainda não é possível encontrar uma documentação mais aprofundada sobre o *framework* e diferentes materiais sobre o uso do *Amazon SageMaker*, o que dificulta o acesso a todos seus recursos de forma mais prática, limitando assim diferentes tipos de análises, métricas e visualização de dados. Ainda assim, com a exploração do *framework*, aliado a básica documentação encontrada, foi essencial para a evolução desta pesquisa.

Por fim, conclui-se nesse trabalho que a plataforma *Amazon SageMaker* apresenta um grande avanço para pesquisas relacionadas com a Visão Computacional. Esta ferramenta acaba abstraindo o processo de configuração de ambientes de desenvolvimento e configuração de algoritmos. Assim, não foram encontradas dificuldades para utilizar o algoritmo *SSD: Single Shot Detector* como arquetipo de arquitetura para a classificação de frutos de acerola com base em sua cor.

6.2 Trabalhos futuros

Espera-se que em trabalhos futuros, obter novas formas para obtenção de dados. Neste trabalho as imagens foram retidadas de vídeos gravados direto do campo, o que ocasionou uma perca já esperada da qualidade das imagens. A perca de qualidade afeta diversos pontos da metodologia proposta, desde a marcação das regiões de interesse até o aprendizado do modelo. A resolução das imagens é fator fundamental para a evolução do aprendizado de modelos de visão computacional.

Assim, analisar a viabilidade da criação de um processo robotizado para obtenção de imagens do campo é fator fundamental para uma maior precisão no processo de detecção e classificação dos frutos. Além disso, continuar a explorar o *Amazon SageMaker* serão os próximos passos desta pesquisa. Durante este trabaho a configuração de hiperparâmetros se manteve estática, pois não ouve tempo hábil para realizar o processo de treinamento com variação dessas configurações. Logo, novos treinamentos devem ser realizados com novos ajuste de hiperparâmetros, que podem potencializar positivamente os resultados obtidos.

REFERÊNCIAS

AMIT, Y.; FELZENSZWALB, P.; GIRSHICK, R. Object detection. **Computer Vision: A Reference Guide**, Springer, p. 1–9, 2020.

BACKES, A. R.; JUNIOR, J. J. d. M. S. **Introdução à visão computacional usando Matlab**. [S.l.]: Alta Books Editora, 2019.

BARGOTI, S.; UNDERWOOD, J. Deep fruit detection in orchards. In: IEEE. **2017 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2017. p. 3626–3633.

BORJI, A.; CHENG, M.-M.; HOU, Q.; JIANG, H.; LI, J. Salient object detection: A survey. **Computational visual media**, Springer, v. 5, n. 2, p. 117–150, 2019.

BORTH, M. R.; IACIA, J. C.; PISTORI, H.; RUVIARO, C. F. A visão computacional no agronegócio: Aplicações e direcionamentos. **7º Encontro Científico de Administração, Economia e Contabilidade (ECAECO)**, 2014.

DRUZHKOV, P.; KUSTIKOVA, V. A survey of deep learning methods and software tools for image classification and object detection. **Pattern Recognition and Image Analysis**, Springer, v. 26, n. 1, p. 9–15, 2016.

FITCH, F. B. Warren s. mcculloch and walter pitts. a logical calculus of the ideas immanent in nervous activity. bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133. **The Journal of Symbolic Logic**, Cambridge University Press, v. 9, n. 2, p. 49–50, 1944.

GLOBO RURAL. **No Ceará, cresce o cultivo da acerola orgânica para exportação**. 2014. Disponível em: <https://g1.globo.com/economia/agronegocios/noticia/2014/03/no-ceara-cresce-o-cultivo-da-acerola-organica-para-exportacao.html>. Acesso em: 07 mar. 2014.

JANIESCH CHRISTIAN E ZSCHECH, P. e. H. K. Aprendizado de máquina e aprendizado profundo. **Mercados Eletrônicos**, v. 31.

KELNER, J. A.; ORECCHIA, L.; SIDFORD, A.; ZHU, Z. A. A simple, combinatorial algorithm for solving sdd systems in nearly-linear time. In: **Proceedings of the forty-fifth annual ACM symposium on Theory of computing**. [S.l.: s.n.], 2013. p. 911–920.

KOIRALA, A.; WALSH, K.; WANG, Z.; MCCARTHY, C. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'mangoyolo'. **Precision Agriculture**, Springer, v. 20, n. 6, p. 1107–1135, 2019.

KOVÁCS, Z. L. **Redes neurais artificiais**. [S.l.]: Editora Livraria da Fisica, 2002.

KOVALEV, V.; KALINOVSKY, A.; LIAUCHUK, V. Deep learning in big image data: Histology image classification for breast cancer diagnosis. In: SN. **Big Data and Advanced Analytics, Proc. 2nd International Conference, BSUIR, Minsk**. [S.l.], 2016. p. 44–53.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LEITE, T. d. M. **Deep learning em dois estágios para detecção e classificaçãode doenças em folhas de plantas com aplicação em dispositivos móveis**. Tese (Doutorado) — Universidade de São Paulo, 2021.

LEITE, W. L. S. Análise de viabilidade do uso de aprendizagem profunda para detecção de frutos de acerola em imagens. 2020.

LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; BERG, A. C. Ssd: Single shot multibox detector. In: SPRINGER. **European conference on computer vision**. [S.l.], 2016. p. 21–37.

MILANO, D. de; HONORATO, L. B. **Visao computacional**. 2014.

MONTANARI, R. **Detecção e classificação de objetos em imagens para rastreamento de veículos**. Tese (Doutorado) — Universidade de São Paulo, 2016.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. **arXiv preprint arXiv:1511.08458**, 2015.

PONTES, A.; SOARES, F.; LIMA, F.; DINIZ, C. Uso do ciclo fenológico da aceroleira para padronização do ponto de colheita mecanizada. In: SN. **XLIV Congresso Brasileiro de Engenharia Agrícola**. [S.l.], 2015.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. **Advances in neural information processing systems**, v. 28, 2015.

RITZINGER, R.; RITZINGER, C. H. S. P. Acerola. **Embrapa Mandioca e Fruticultura-Artigo em periódico indexado (ALICE)**, Informe Agropecuário, Belo Horizonte, MG: EPAMIG, v. 32, n. 264, p. 17-25 . . . , 2011.

SAMANTHA CERQUETANI. **Acerola fortalece a imunidade; veja 9 benefícios da fruta tropical**. 2021. Disponível em: <https://www.uol.com.br/vivabem/noticias/redacao/2021/06/21/acerola-fortalece-a-imunidade-veja-9-beneficios-da-fruta-tropical.htm?next=0004H57U45N>. Acesso em: 21 jun. 2021.

TEUWEN, J.; MORIAKOV, N. Convolutional neural networks. In: **Handbook of medical image computing and computer assisted intervention**. [S.l.]: Elsevier, 2020. p. 481–501.