

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA**

MARCOS VINÍCIUS LEAL DA SILVA

**CONTAGEM DE LARANJAS UTILIZANDO REDE
NEURAL CONVOLUCIONAL**

Florianópolis, 2022

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA**

MARCOS VINÍCIUS LEAL DA SILVA

CONTAGEM DE LARANJAS UTILIZANDO REDE NEURAL CONVOLUCIONAL

Trabalho de conclusão de curso submetido
ao Instituto Federal de Educação, Ciência
e Tecnologia de Santa Catarina como parte
dos requisitos para obtenção do título de
engenheiro eletrônico

Orientador:
Prof. Me. Samir Bonho

Florianópolis, 2022

Ficha de identificação da obra elaborada pelo autor.

Silva, Marcos Vinícius Leal da
Contagem de Laranjas Utilizando Rede Neural Convolucional
/ Marcos Vinícius Leal da Silva; orientação de Samir
Bonho. - Florianópolis, SC, 2022.
60 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Eletrônica. Departamento Acadêmico
de Eletrônica.
Inclui Referências.

1. Rede Neural Convolucional. 2. Mask R-CNN. 3.
Detectron2. 4. Contagem de laranjas. 5. Segmentação de
objetos. I. Bonho, Samir. II. Instituto Federal de Santa
Catarina. III. Contagem de Laranjas Utilizando Rede
Neural Convolucional.

CONTAGEM DE LARANJAS UTILIZANDO REDE NEURAL CONVOLUCIONAL

MARCOS VINÍCIUS LEAL DA SILVA

Este Trabalho foi julgado adequado para obtenção do Título de Engenheiro Eletrônico em dezembro de 2022 e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Eletrônica do Instituto Federal de Educação Ciência, e Tecnologia de Santa Catarina.

Florianópolis, 21 de dezembro, 2022.

Banca Examinadora:

Documento assinado digitalmente
 SAMIR BONHO
Data: 27/02/2023 16:18:11-0300
Verifique em <https://verificador.itl.br>

 Samir Bonho, Me.
Documento assinado digitalmente
PEDRO GIASSI JUNIOR
Data: 14/02/2023 16:50:02-0300
CPF: 219.227.438-60
Verifique as assinaturas em <https://v.ifsc.edu.br>

 Pedro Giassi Junior, Dr.
Documento assinado digitalmente
REGINALDO STEINBACH
Data: 14/02/2023 11:44:58-0300
Verifique em <https://verificador.itl.br>

Reginaldo Steinbach, Me.

*Este trabalho é dedicado à minha família,
principalmente minha mãe, pai, irmãos e irmã
pois mesmo quando eu achava que não conseguiria,
eles mesmo assim confiaram em mim.*

AGRADECIMENTOS

Gostaria de agradecer à todos que me incentivaram de forma direta ou indireta em continuar a caminhada.

Em especial gostaria de agradecer aos meus pais, Irimar e Dalva, por terem me gerado e educado. Por todo carinho, zelo, diálogos e uma base sólida que fizeram ser quem sou hoje. Obrigado por me proporcionarem tudo que estava no alcance de vocês para que eu pudesse estudar, vocês sempre me apoiaram em todos os sentidos e não me deixaram desistir dos meus sonhos.

Obrigado aos meus irmãos e irmã, Willian, Wilson e Suéllem por toda paciência em tempos bons e ruins. Vocês me inspiraram e inspiram a ser uma pessoa melhor a cada dia, além de aprender com vocês que deve-se dar um passo de cada vez.

Willian, de coração imenso e generoso, desde pequeno eu via você estudando e se dedicando cada vez mais para atingir seus objetivos. Os tropeços no caminho nunca foram motivos para te fazer baixar a cabeça, pelo contrário, davam mais combustível para buscar mais e mais.

Wilson, de mente brilhante e dedicada, apesar de brigar contigo quando pequeno e te encher a paciência na sua graduação, suas conquistas são frutos de toda dedicação. Nunca vou me esquecer das conversas que tivemos quando os tempos ruins chegaram na minha caminhada acadêmica, obrigado por todas as palavras e dicas.

Suéllem, de olhar aventureiro e palavras sinceras, pela idade mais próxima sempre brincamos e nos entedemos muito bem, saiba que você me ensinou muito sobre como enxergar as situações sob diferentes perspectivas. Sua jornada é fonte de inspiração e muito obrigado por compartilhar suas palavras sinceras nos meus momentos difíceis, você é sensacional!

Meu muito obrigado aos companheiros da jornada acadêmica, em especial ao PK, Sérgio, Belisário, Heloíza e Suzi. Com cada um de vocês para estudar, fazer trabalhos e projetos juntos a caminhada foi difícil, mas recheadas de boas gargalhadas. Obrigado por me aturarem nos melhores e principalmente nos piores momentos da academia, vocês são pessoas incríveis.

Agradeço também a Isabela Franco Dantas por todo apoio e incentivo quando nem eu mesmo tinha forças para continuar, você me faz muito bem e sabe disso. Obrigado a dona Claudinha e toda família que me acolheu e me impulsionou a correr atrás dos meus sonhos, além das limpezas e orações.

Gostaria de agradecer também aos professores Leandro Schwarz e Jony Laureano Silveira, ambos meus orientadores de bolsa de pesquisa. Muito do profissional

que sou hoje é fruto dos ensinamentos, puxões de orelha sempre merecidos e longas conversas na iniciação científica. Meu muito obrigado por toda paciência, dedicação e diálogos comigo.

Por fim agradeço aos professores do IFSC que me guiaram nessa jornada da graduação, em especial ao professor Samir Bonho por todo apoio nessa reta final de defesa do TCC. Muito obrigado por confiar em mim e no desenvolvimento do trabalho, com seus empurrões e palavras de motivação.

RESUMO

Realizar a contagem de laranjas em lavouras cítricas é uma automação importante para escalabilidade e faturamento de uma lavoura. À medida que novas tecnologias são desenvolvidas, as redes neurais convolucionais tem ganhado destaque na detecção e segmentação de objetos, e por conta disso, diferentes arquiteturas são propostas em busca de proporcionar alta confiabilidade nessa tarefa. Dessa forma, propõe-se nesse trabalho a comparação de duas redes neurais convolucionais distintas, utilizando uma arquitetura de Mask R-CNN e pré-treinadas sob o dataset COCO, na tarefa específica de contagem de laranjas na laranjeira. Utilizou-se um ground truth, feito por inspeção humana, de com imagens contendo 33 resoluções distintas para o teste. Os algoritmos foram desenvolvidos no ambiente Google Colab, sendo ambos códigos disponibilizados para reprodução, estudos, melhorias e trabalhos futuros. O desempenho alcançado foi de uma acurácia média de 55,85 com Mask R-CNN e 56,04 com Detectron2.

Palavras-chave: Rede Neural Convolucional. Mask R-CNN. Detectron2. Contagem de Laranjas. Segmentação de objetos.

ABSTRACT

Counting oranges in citrus crops is an important automation for scalability and billing of a tillage. As new technologies are developed, convolutional neural networks have gained prominence in object detection and segmentation, and because of this, different architectures are proposed in order to provide high reliability in this task. Thus, this work proposes the comparison of two distinct convolutional neural networks, using a Mask R-CNN architecture and pre-trained under the COCO dataset, in the specific task of counting oranges on the orange tree. A ground truth was used, made by human inspection, with images containing 33 different resolutions for the test. The algorithms were developed in the Google Colab environment, both codes being available for reproduction, studies, improvements and future works. The performance achieved was an average accuracy of 55.85% with Mask R-CNN and 56.04% with Detectron2.

Keywords: Convolutional Neural Network. Mask R-CNN. Detectron2. Counting Oranges. Object segmentation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Comparativo entre a percepção humana com a computacional. . . .	17
Figura 2 – Relação entre Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo.	17
Figura 3 – Rede Neural Convolucional e Visão Computacional.	19
Figura 4 – Visão geral da arquitetura de uma CNN.	19
Figura 5 – Modelagem em uma matriz tridimensional de uma imagem em RGB. . . .	20
Figura 6 – Operação de convolução na camada de extração de recursos. . . .	21
Figura 7 – <i>Padding</i> aplicado nas bordas da imagem de entrada.	21
Figura 8 – Função de ativação ReLU.	22
Figura 9 – Função de ativação ReLU aplicada.	22
Figura 10 – <i>Average pooling</i> e <i>max pooling</i> aplicados em uma matriz 4 x 4. . . .	23
Figura 11 – Camada de Classificação.	24
Figura 12 – Camada de <i>Softmax</i>	25
Figura 13 – Exemplo de reconhecimento, detecção e segmentação semântica de objetos.	26
Figura 14 – Panorama das CNNs.	27
Figura 15 – Estrutura da Mask R-CNN.	28
Figura 16 – Estrutura detalhada de uma Mask R-CNN.	29
Figura 17 – Rede <i>Backbone</i> simplificada.	29
Figura 18 – Rede RPN exibindo 49 âncoras em uma imagem.	30
Figura 19 – Regiões de Interesse (RoI) gerados pela rede RPN.	31
Figura 20 – Classificador de RoI e Regressor de caixa delimitadora.	32
Figura 21 – Máscara de segmentação gerada na Mask R-CNN.	32
Figura 22 – Diferentes tipos de detecções feitas com Detectron2.	33
Figura 23 – Segmentação panóptica feita com Detectron2.	34
Figura 24 – Diagrama de validação das redes pré-treinadas sob o mesmo <i>dataset</i>	35
Figura 25 – Laranja do <i>ground truth</i> , contagem manual de 42 laranjas.	36
Figura 26 – Laranja do <i>ground truth</i> , contagem manual de 68 laranjas.	37
Figura 27 – Laranja do <i>ground truth</i> , contagem manual de 16 laranjas.	37
Figura 28 – Fluxograma do algoritmo desenvolvido aplicando a rede neural Mask R-CNN.	40
Figura 29 – Laranja utilizada para teste do algoritmo com inspeção visual de 5 laranjas.	41
Figura 30 – Laranja após algoritmo de segmentação de imagens com Mask R-CNN.	41
Figura 31 – Fluxograma do algoritmo desenvolvido aplicando a rede neural Detectron2	45

Figura 32 – Laranjeira após algoritmo de segmentação de imagens com Detectron2.	46
Figura 33 – Imagem com resolução de 219 x 183 pixels.	48
Figura 34 – Segmentação com Mark R-CNN.	48
Figura 35 – Segmentação com Detectron2.	49
Figura 36 – Imagem com resolução de 2560 x 1707 pixels.	49
Figura 37 – Segmentação com Mask R-CNN.	50
Figura 38 – Segmentação com Detectron2.	50
Figura 39 – Segmentação de uma maçã onde não há.	51
Figura 40 – Segmentação de uma cenoura onde não há	52
Figura 41 – Segmentação com Detectron2.	53
Figura 42 – Segmentação de duas laranjas distintas como sendo somente uma.	54
Figura 43 – Segmentação de uma laranja como duas pela sobreposição de folhas.	55
Figura 44 – Segmentação de uma laranja como duas pela sobreposição da árvore.	56
Figura 45 – Dificuldade de segmentação pela sobreposição da folhas.	57

LISTA DE TABELAS

Tabela 1 – Exemplo de cálculo detalhado de <i>softmax</i>	24
Tabela 2 – Contagem de laranja da rede neural Mask R-CNN e acurácia individual.	43
Tabela 3 – Contagem de laranja da rede neural utilizando Detectron2 e acurácia individual	47

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Justificativa	14
1.2	Descrição do problema	15
1.3	Objetivo geral	15
1.4	Objetivos específicos	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Visão Computacional	16
2.2	Redes Neurais Convolucionais	18
2.2.1	Arquitetura Padrão de Redes Neurais Convolucionais	19
2.2.2	Camada de Entrada	20
2.2.3	Camada de Extração de Recursos	20
2.2.4	Camada de Classificação (Saída)	23
2.3	Arquiteturas de redes neurais utilizadas em segmentação de imagem	25
2.3.1	Mask R-CNN	28
2.3.2	Detectron2	32
3	DESENVOLVIMENTO	35
3.1	Levantando o <i>Ground Truth</i>	35
3.2	Métricas utilizadas para comparação das redes neurais	37
3.3	Algoritmo utilizado para rede neural Mask R-CNN	39
3.4	Aplicando a rede neural Mask R-CNN pré-treinada com MS COCO dataset no <i>Ground Truth</i>	42
3.5	Algoritmo utilizado para rede neural Detectron2	44
3.6	Aplicando a rede neural Detectron2 pré-treinada com MS COCO dataset no <i>Ground Truth</i>	46
4	DISCUSSÃO	48
5	CONSIDERAÇÕES FINAIS	58
	REFERÊNCIAS	60

1 INTRODUÇÃO

A população mundial já ultrapassou 7,7 bilhões de habitantes em 2020, segundo estimativa feita pelas Nações Unidas (NATIONS, 2021). Calcula-se que a população mundial atinja 9 bilhões em 2050, sendo que os principais desafios para a humanidade serão energia, água, alimentos, ambiente e pobreza. (MASSRUHÁ; LEITE, 2017). Para produção de alimentos são necessários recursos finitos e de grande valor, dentre eles terra, água e energia. O tempo para produção e distribuição dos alimentos também precisa ser levado em conta, haja vista a complexidade de todo processo, que compreende desde o plantio até a chegada ao prato de cada habitante. Portanto a utilização de maneira planejada e consciente desses recursos tem sido imprescindível para ampliar a quantidade e a qualidade do alimento que é produzido.

Neste sentido, a agricultura está em constante evolução para trazer as melhores técnicas de plantio, colheita, armazenamento e distribuição. O atual estágio é denominado agricultura 4.0. Segundo Lisbinski et al. (2020):

"... a Agricultura 4.0 consiste na conectividade da rede interna e externa das operações agrícolas e na integração dessas informações em uma mesma plataforma, possibilitando a tomada de decisão em tempo real."

A agricultura 4.0 surge de mãos dadas com a indústria 4.0. Nela os equipamentos e sistemas são conectados na internet para coleta, transmissão e processamento de dados em tempo real, auxiliando assim a tomada de decisão. Estas decisões são embasadas em dados, diferindo da metodologia feita no passado, período em que a experiência e intuição do produtor predominavam na tomada de ação.

O potencial atual de replicar e escalar a produção das lavouras é forte aliado para superar os desafios apresentados. Nesse mar de dados que os equipamentos e sistemas geram, a inteligência artificial e o aprendizado de máquina são poderosas ferramentas para extrair informações dos dados coletados e auxiliar o produtor na tomada de decisão. Segundo Fernando Martins, conselheiro de empresas de tecnologia voltadas ao agronegócio, estima-se que no futuro o agricultor fará um acompanhamento, monitoramento e apoio aos processos já em curso da lavoura (ZAPAROLLI, 2020). Esses processos seriam então mapeados previamente com o auxílio das ferramentas comentadas e, em seguida, automatizados, tornando assim a lavoura mais independente. Por exemplo, a rota de um trator para colher determinado fruto pode ser otimizada com base na posição GPS do veículo e de áreas que já podem ser colhidas, economizando assim o consumo de combustível.

A qualidade do fruto é importante no faturamento da lavoura frutífera, pois um fruto sem defeitos consegue agregar valor no produto. Logo essa característica também é abordada nos trabalhos, como mostra Rong, Rao e Ying (2017), que utilizam visão computacional para detectar defeitos nas superfícies de laranjas. O algoritmo proposto

utiliza uma segmentação local de janela de comparação e foi testado com 1191 amostras de laranjas, tendo detectado 97% dos defeitos na superfície da laranja. No trabalho os pesquisadores também detalham o procedimento de processamento de imagem utilizado, sendo remoção de pixels de fundo, binarização utilizando segmentação local, subtração e modificação morfológica.

Outro dado que auxilia nos processos de lavouras frutíferas é a estimativa de produção, a contagem de frutos, impactando diretamente no faturamento. Na Austrália, por exemplo, Payne et al. (2013) utilizaram o método de segmentação de imagem para estimar o rendimento de uma safra de manga. Esse algoritmo em questão apresentou uma correlação alta ($R^2 = 0,91$) entre a quantidade de frutos contados de forma autônoma e manual, sendo imagens tiradas de quatro lados de árvores individuais. Todavia, o mesmo algoritmo teve baixa eficácia à medida que o número de frutos nas árvores aumentou, sendo testado em um grupo de 555 árvores com imagens tiradas em um lado de cada árvore somente. Payne et al. ainda ressaltou os problemas de incidência direta de luz na captura das imagens, prejudicando o desempenho do algoritmo proposto.

Na China, Wang et al. (2017), testaram a eficácia de um algoritmo robusto de segmentação de frutas em imagens capturadas em ambientes naturais com a variação de iluminação para sistemas de visão em robôs de colheita. O método utilizado no algoritmo consiste na aplicação de uma transformada wavelet melhorada nas imagens de frutas para normalizar a iluminação na superfície do objeto. Neste trabalho foram utilizadas imagens capturadas em dias ensolarados e nublados de três tipos de frutas com cores diferentes, onde o algoritmo proposto conseguiu lidar com a variação de iluminação nas frutas e segmentar precisamente os três tipos de frutas. A taxa de acerto do algoritmo em dias ensolarados foi de 95,3%, 97,2% e 89,2% para detecção de lichia, uva e citrino respectivamente. Já em dias nublados a taxa diminuiu um pouco, sendo 91,6%, 93,6% e 87,3% de modo recíproco. O tempo médio de processamento do algoritmo foi de 1365 ms, sendo viável para aplicações em tempo real.

Além dos trabalhos citados, existem inúmeros estudos e uma grande linha de pesquisa com relação à contagem de frutos, utilizando técnicas de aprendizado de máquinas. É nesta linha que o presente trabalho irá se estender, aplicando arquiteturas de redes neurais convolucionais para classificação e contagem de frutas, no caso, laranjas.

1.1 Justificativa

A estimativa de produção de lavouras frutíferas é um processo que impacta diretamente no faturamento da mesma, sendo um trabalho demorado e penoso se feito de forma manual. Em lavouras muito grandes estimativas são feitas contando os frutos

de algumas árvores e fazendo uma média, todavia é possível acelerar e melhorar a estimativa com métodos computacionais.

Algoritmos de processamento de imagem e a utilização de redes neurais convolucionais permitem o aprimoramento da estimativa de produção, acelerando o processo de contagem de frutos podendo auxiliar também na organização da colheita.

1.2 Descrição do problema

A estimativa de produção de uma lavoura frutífera baseia-se no histórico do plantio pelo agricultor e na experiência do mesmo. Os dados podem ser compartilhados entre os agricultores, todavia pouca informação se tem sobre a produção entre o plantio e a colheita. O acompanhamento da produção de uma lavoura é um trabalho muitas vezes visual, repetitivo e que escala com o tamanho da mesma. Dessa maneira, a problemática que orienta esse Trabalho de Conclusão de Curso é: Como utilizar redes neurais convolucionais para estimar a produção de uma lavoura frutífera.

1.3 Objetivo geral

Este trabalho tem como objetivo testar duas arquiteturas de redes neurais convolucionais para contagem de laranjas.

1.4 Objetivos específicos

Com o objetivo geral apresentado, destacam-se os seguintes objetivos específicos:

- a) estudar as etapas de segmentação e extração de informação no processamento de imagens;
- b) realizar uma pesquisa bibliográfica referente aos algoritmos de detecção de objetos em imagens;
- c) estudar e sintetizar o conceito de aprendizado de máquina, abordando modelos que serão utilizados no escopo do trabalho;
- d) utilizar os modelos pré-treinados escolhidos para detecção do objeto escolhido;
- e) comparar o desempenho dos diferentes modelos de aprendizado de máquina escolhidos para detecção do objeto em imagens estáticas (fotos);

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar uma síntese de visão computacional, além de explorar a arquitetura padrão de rede neurais convolucionais. Em seguida, será apresentado também a rede neural convolucional escolhida para realizar a segmentação das laranjas.

2.1 Visão Computacional

Visão computacional é um conceito amplo, sendo em resumo a transformação de um dado estático, como uma foto, ou dinâmico, como um vídeo, em uma decisão ou uma nova representação (KAEHLER; BRADSKI, 2017). Os computadores precisam que um sensor faça a captura do dado para posteriormente poder processá-los. O que o computador entende são números, logo o sensor faz a conversão elétrica da captura da imagem em um valor de tensão, por exemplo, que indica a intensidade de uma cor. A Figura 1 exibe um comparativo da percepção humana e do que um sistema computacional recebe como entrada de dados.

A diferença principal não é na forma em que os dados estão exibidos em si, mas em como nós enxergamos um carro enquanto uma máquina enxerga um mar de números. Nós aprendemos a identificar um carro, nos foi ensinado no decorrer de anos a processar as informações à nossa volta e com base nesse contexto saber identificar um objeto, uma cor, um rosto, um animal, e assim por diante. Muito desse conhecimento é subjetivo e intuitivo, portanto difícil de articular de forma formal (GOODFELLOW; BENGIO; COURVILLE, 2016). Os sistemas computacionais precisam aprender a identificar um objeto com base na extração de características a partir de um determinado conjunto de dados. Na maior parte dos casos isto não é uma tarefa fácil. Uma ferramenta muito utilizada para achar associações e correlações entre o mar de números que representam as imagens a serem classificadas é o aprendizado de máquinas (em inglês, *machine learning*).

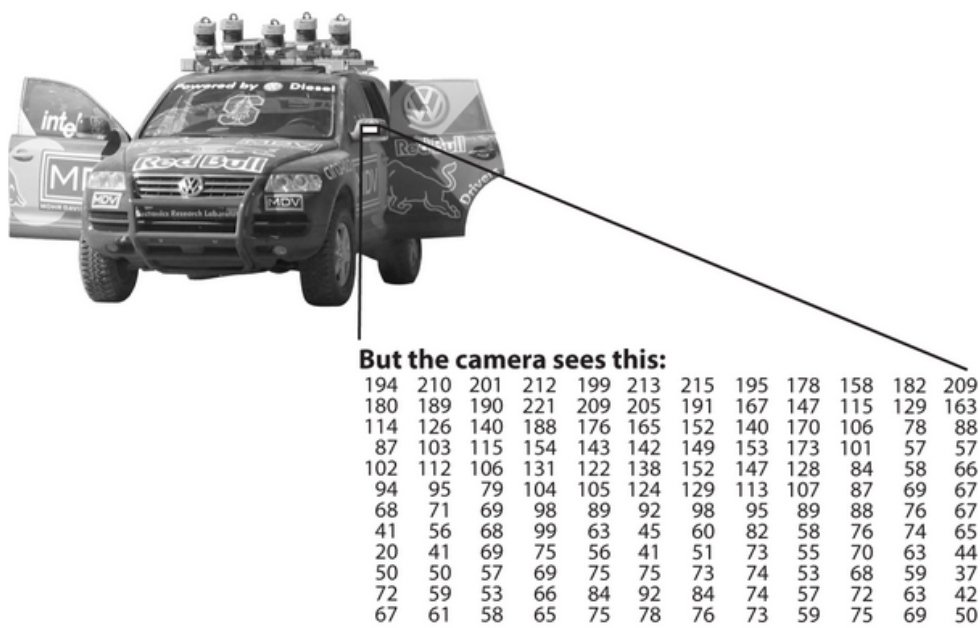
O aprendizado de máquina tem se tornado objeto de estudo com o avanço do poder computacional no decorrer dos anos. Segundo Samuel (1959), pioneiro de inteligência artificial na IBM e em Stanford, o aprendizado de máquina é definido como:

“Campo de estudo que proporciona aos computadores a habilidade de aprender sem serem explicitamente programados.”

O aprendizado profundo, é um campo de estudo dentro do aprendizado de máquina, que por sua vez está contido dentro da inteligência artificial. A Figura 2 exibe a relação entre os três conceitos mencionados.

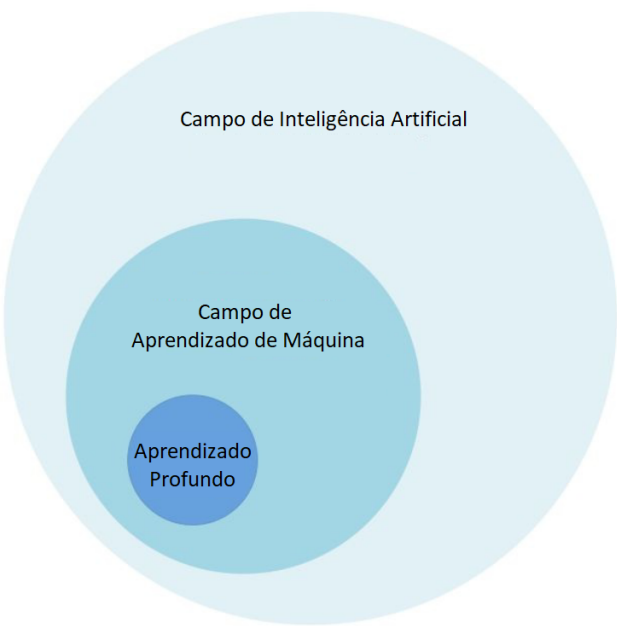
Como técnicas de aprendizado profundo, pode-se destacar as redes neurais artificiais. Porém, no campo de visão computacional, são as redes neurais convolucio-

Figura 1 – Comparativo entre a percepção humana com a computacional.



Fonte: (KAEHLER; BRADSKI, 2017).

Figura 2 – Relação entre Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo.



Fonte: Adaptado de Patterson e Gibson (2017).

nais que predominam nas tarefas de segmentação de imagens.

2.2 Redes Neurais Convolucionais

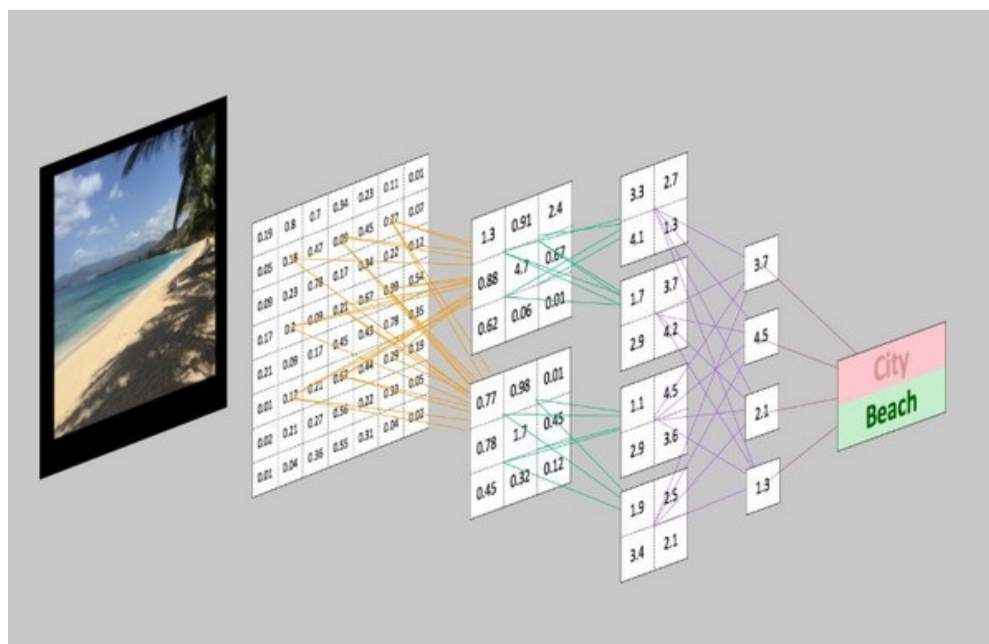
A abordagem convencional de programação é detalhar para a máquina o que deve ser feito, dividindo o problema em tarefas menores. Uma rede neural, por sua vez, utiliza dados observacionais para aprender como resolver o problema (NIELSEN, 2018).

As redes neurais antigas consumiam muito recurso de processamento. Com o avanço da tecnologia e mudanças nos modelos arquitetônicos das redes neurais, foi possível avançar nas aplicações desse campo. As mudanças mencionadas vão desde: maior quantidade de neurônios ao comparar com redes anteriores, maior complexidade nas conexões entre camadas, melhor eficiência e equilíbrio na exploração dos recursos computacionais no decorrer do treinamento da rede e da extração de dados (PATTERSON; GIBSON, 2017).

Quando uma rede neural possui mais de duas camadas ela é definida como uma rede neural profunda. Existem diversas arquiteturas de redes neurais profundas, dentre elas destacam-se: Redes Neurais Pré Treinadas Sem Supervisão (UPNs - Unsupervised Pretrained Networks); Redes Neurais Convolucionais (CNNs - Convolutional Neural Networks); Redes Neurais Recorrentes (Recurrent Neural Networks); Redes Neurais Recursivas (Recursive Neural Networks) (PATTERSON; GIBSON, 2017).

As redes neurais convolucionais, também conhecidas como CNNs, são muito utilizadas no reconhecimento de objetos em imagens e estão consistentemente no topo de competições de classificação de imagens. Como o próprio nome já indica, a partir da função matemática convolução, que consiste na multiplicação e soma de pixels, as CNNs conseguem aprender recursos de ordem superior nos dados. Elas são capazes de identificar diversos objetos como frutas, animais, placas e outros aspectos visuais, além de serem boas para análise de som e reconhecimento de caracteres em texto.

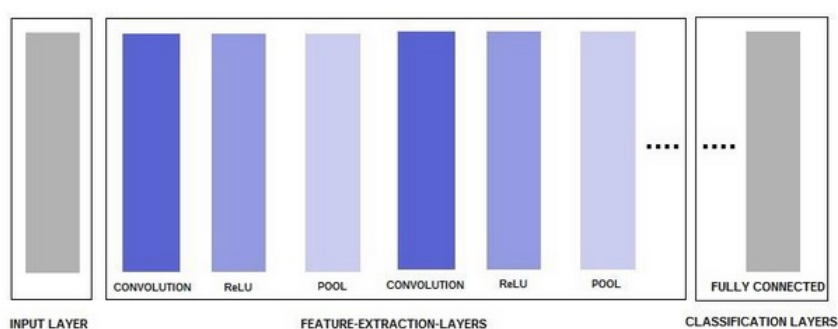
A Figura 3 exibe a imagem de uma praia, representada por uma matriz normalizada de 8 x 7 pixels (altura x largura), onde todos os pixels estão conectados à camada de convolução, de tamanho 3 x 3. Essa camada de convolução é a responsável por varrer a imagem e extrair os recursos, multiplicando e somando os pixels. Com base nesses dados é possível extrair a informação que a imagem trata-se de uma praia, e não de uma cidade.

Figura 3 – Rede Neural Convolucional e Visão Computacional.

Fonte: (PATTERSON; GIBSON, 2017).

2.2.1 Arquitetura Padrão de Redes Neurais Convolucionais

As CNNs transformam os dados de entrada da camada de entrada através de todas as camadas conectadas em um conjunto de pontuações de classe fornecidas pela camada de saída (PATTERSON; GIBSON, 2017). Existem muitas variações da arquitetura CNN, mas eles são baseados no padrão de camadas, conforme exibido na Figura 4.

Figura 4 – Visão geral da arquitetura de uma CNN.

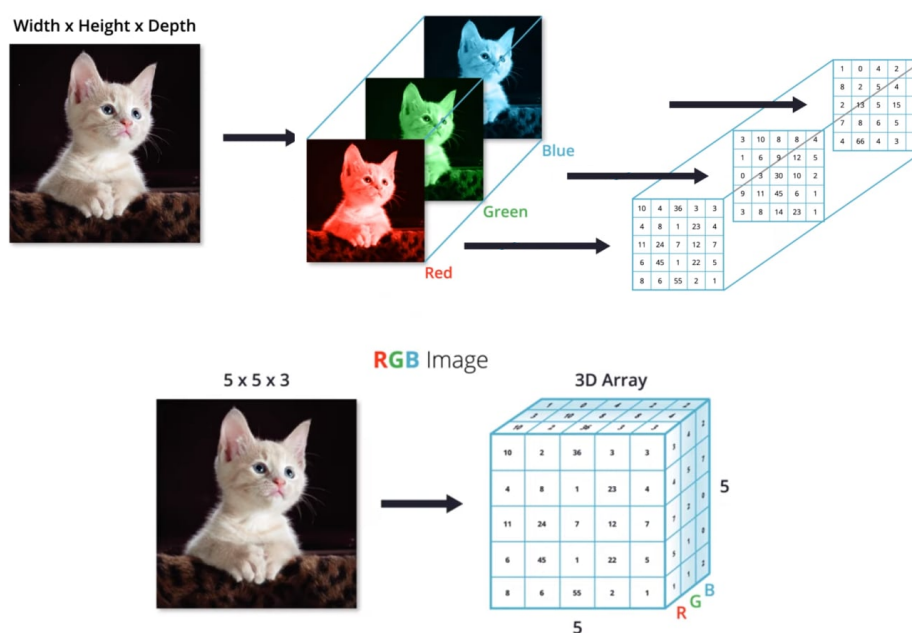
Fonte: (PATTERSON; GIBSON, 2017).

A seguir serão detalhadas cada uma das camadas exibidas e suas características. Cada camada transforma o volume de entrada 3D da camada anterior em um volume de saída 3D de ativação de neurônios com alguma função diferenciável que pode ou não ter parâmetros.

2.2.2 Camada de Entrada

Tratando-se de imagens, a camada de entrada consiste em uma estrutura de neurônios tridimensionais que contém os elementos largura, altura e profundidade. Essa última corresponde aos canais RGB (do inglês Red, Green, Blue), sendo o sistema de cores primárias. A Figura 5 exibe a representação de como uma imagem em RGB é modelada em uma matriz tridimensional.

Figura 5 – Modelagem em uma matriz tridimensional de uma imagem em RGB.



Fonte: Adaptado de Balachandran (2020).

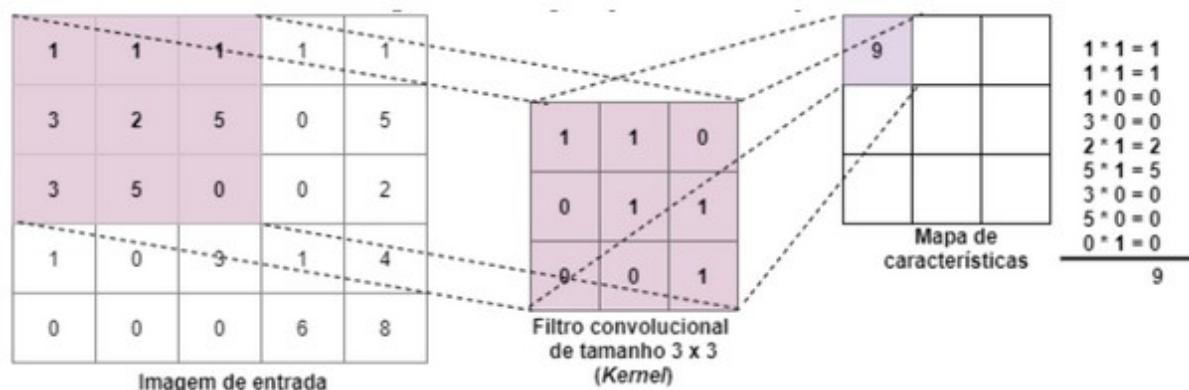
2.2.3 Camada de Extração de Recursos

Para a extração de recursos utiliza-se uma repetição sequencial de: (1) Convolução - que aplica a multiplicação e soma de pixels; (2) ReLU - sendo uma função de ativação; (3) Pooling (agrupamento) - responsável por obter recursos na imagem e montar sucessivamente recursos de ordem maior.

A camada de convolução recebe os dados de entrada e aplica um filtro de convolução, tendo como resultado um mapa de características na saída (PATTERSON; GIBSON, 2017). É importante ressaltar que ocorre uma varredura na imagem, onde filtros convolucionais são aplicados realizando a multiplicação e soma de pixels. Esses filtros, também conhecidos como kernels, possuem os pesos atribuídos às conexões de um neurônio e são representados por uma matriz. Utilizando então filtros com configurações e tamanhos variados é possível que uma rede neural aprenda diferentes tipos de características de entrada.

Um exemplo da operação realizada na camada de convolução está ilustrada na Figura 6, onde a partir de uma entrada de tamanho 5 x 5 e um kernel de tamanho 3, obtém-se um volume de extração de recursos na saída, representado pelo valor 9.

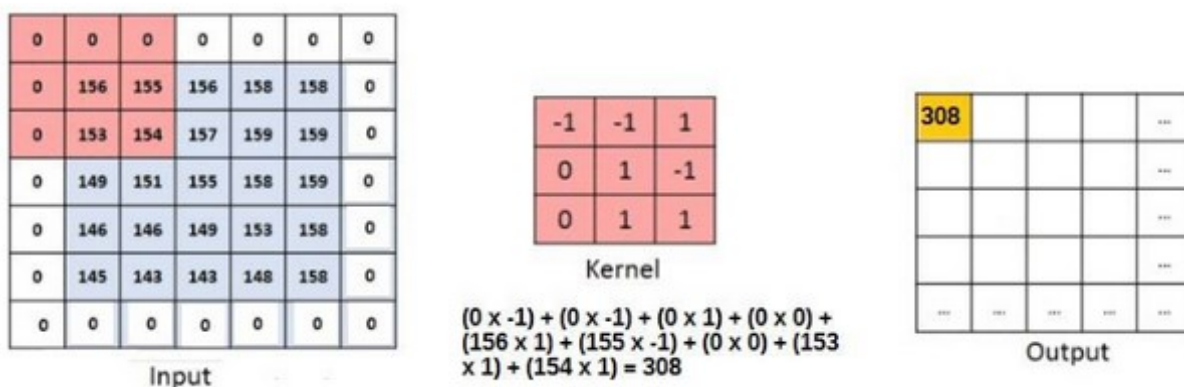
Figura 6 – Operação de convolução na camada de extração de recursos.



Fonte: Adaptado de Patterson e Gibson (2017).

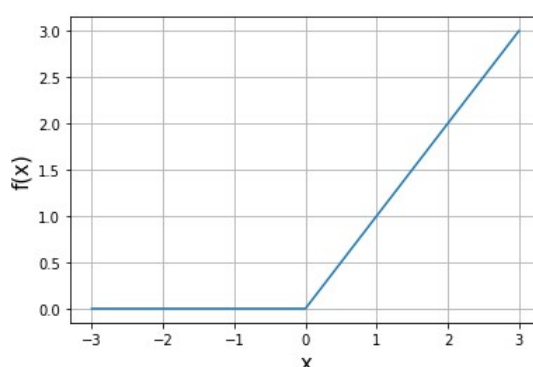
Nota-se que o tamanho da saída da Figura 6 é menor que o tamanho dos dados de entrada, sendo 3 x 3 e 5 x 5 respectivamente. Para controle do tamanho da saída e do filtro convolucional que será aplicado utiliza-se uma técnica chamada de padding. Ela adiciona zeros em determinados pixels da imagem, conforme exibido na Figura 7.

Figura 7 – Padding aplicado nas bordas da imagem de entrada.



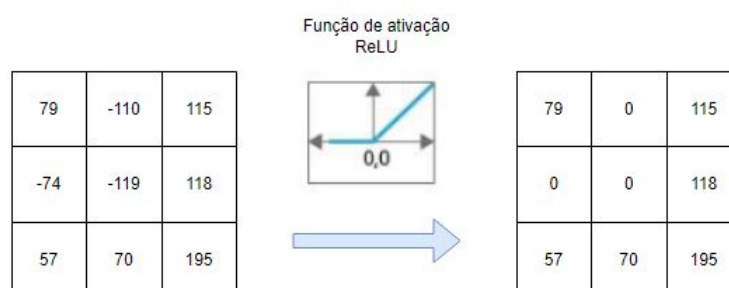
Fonte: Adaptado de Alves (2018).

Logo após a convolução tem-se a função de ativação Unidade Linear Retificada, também conhecida como ReLU, do inglês Rectified Linear Unity. O resultado dessa função ReLU será 0 se a entrada for menor ou igual 0, caso contrário o resultado será o próprio valor de entrada, conforme mostra a Figura 8.

Figura 8 – Função de ativação ReLU.

Fonte: Elaboração própria (2022).

A função dela é alterar os valores dos pixels sem modificar o tamanho da matriz, ou seja, as dimensões da matriz de entrada e saída são as mesmas. A ReLU viabiliza modelar uma distribuição mais contínua para os dados de entrada (PATTERSON; GIBSON, 2017). Um exemplo da função de ativação ReLU aplicada em uma matriz bidimensional é exibida na Figura 9.

Figura 9 – Função de ativação ReLU aplicada.

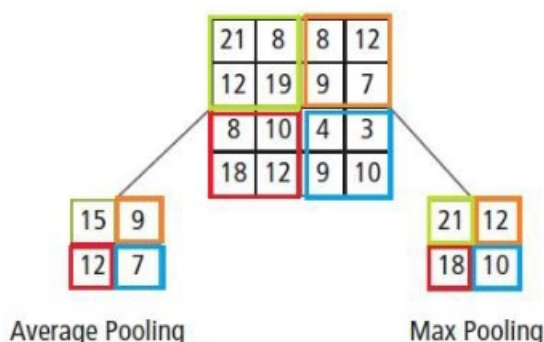
Fonte: Elaboração própria (2022).

Por fim, no bloco que se repete sequencialmente, tem-se o pooling. Ele tem como função reduzir o tamanho espacial (altura x largura) dos dados, diminuindo assim o overfitting. Por realizar essa redução dos recursos, essa camada também é chamada de subamostragem.

O overfitting é a tendência de uma inteligência artificial ficar extremamente boa em reconhecer o padrão de treinamento, portanto tendo um desempenho ruim com dados nunca vistos previamente. Na prática o overfitting ocorre quando o modelo encontra correlações entre os dados de teste de forma randômica que não beneficiarão o modelo ao se deparar com dados não vistos (PATTERSON; GIBSON, 2017). Então imagine uma máquina que foi modelada para reconhecer rostos humanos de forma geral, mas só consegue reconhecer o rosto de uma pessoa específica.

O pooling pode ser implementado por diferentes funções matemáticas, dentre elas duas formas se destacam, sendo average pooling e max pooling. A primeira realiza uma média entre os dados de entrada. Já a segunda faz a seleção do maior valor num conjunto de dados. Por exemplo, pode-se aplicar um dos métodos em uma matriz 4 x 4, dividindo-a em 4 matrizes de 2 x 2 por meio de um kernel de 2 x 2. De cada uma dessas 4 sub-matrizes, resultarão 4 resultados que farão a composição da matriz de pooling. Ambas formas de average pooling e max pooling estão exemplificadas na Figura 10.

Figura 10 – Average pooling e max pooling aplicados em uma matriz 4 x 4.



Fonte: (HIJAZI; KUMAR; ROWEN, 2015).

Dessa forma, a camada de pooling consegue diminuir eventuais ruídos e distorções oriundas da camada anterior.

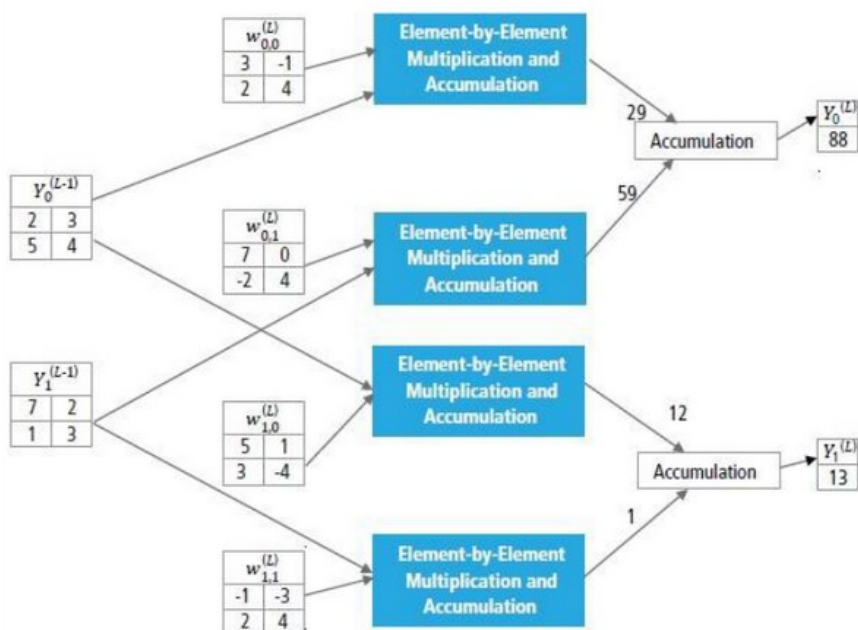
2.2.4 Camada de Classificação (Saída)

Como última etapa de uma rede neural convolucional padrão está a camada de classificação, podendo ser uma ou mais camadas. Elas são totalmente conectadas com a camada anterior, ou seja, essa camada tem uma conexão entre todos os seus neurônios e todos os neurônios da camada anterior.

A Figura 11 exibe a camada de classificação típica de uma rede neural convolucional. Nessa camada ocorre uma soma da camada anterior ($L - 1$) ponderada com os pesos da camada totalmente conectada (L), determinando assim a classe a qual pertence aquela informação. Como resultado então tem-se como saída as probabilidades de correspondência entre as classes definidas na entrada. Essa camada de classificação possui tipicamente uma saída com tamanho $1 \times 1 \times N$, onde N é o número de classes que a informação de entrada está sendo avaliada (PATTERSON; GIBSON, 2017).

Na saída da camada de classificação, geralmente há uma normalização entre a informação de camada totalmente conectada e a quantidade total de possíveis

Figura 11 – Camada de Classificação.



Fonte: (HIJAZI; KUMAR; ROWEN, 2015).

classes. O cálculo de probabilidade para cada classe de saída pode ser feito conforme Equação 2.1, também conhecida como função softmax :

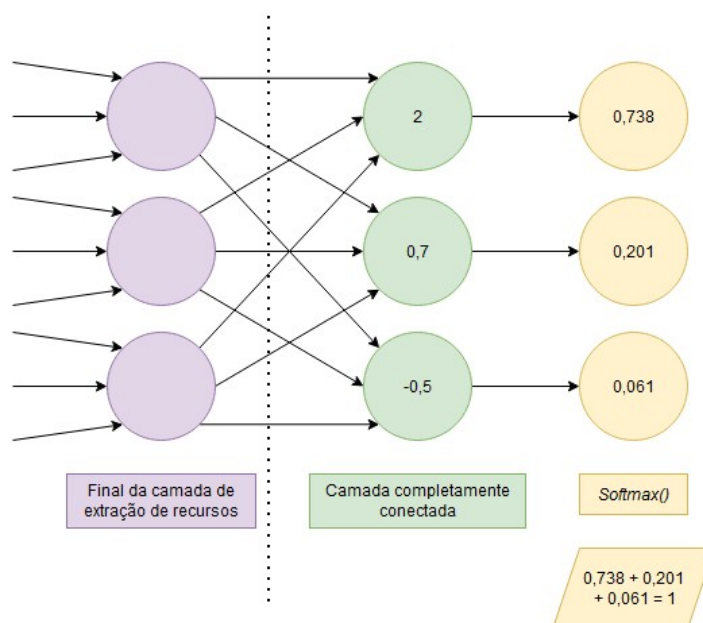
$$\text{softmax}(y_i) = \frac{e^{y_i}}{\sum e^{y_j}} \quad (2.1)$$

Onde o numerador representa a exponencial natural da soma ponderada resultante de um dos neurônios da camada totalmente conectada, e o denominador representa o somatório da exponencial natural de todos os neurônios da camada totalmente conectada. Como resultado tem-se um número entre 0 e 1, sendo o somatório de todas as saídas softmax igual a 1, conforme detalhado na Tabela 1 e exibido na Figura 12

Tabela 1 – Exemplo de cálculo detalhado de *softmax*.

yi	exp(yi)	softmax(yi)
2	7,389	0,738
0,7	2,014	0,201
-0,5	0,607	0,061
$\sum$	10,009	1,000

Fonte: Elaboração própria (2022).

Figura 12 – Camada de *Softmax*.

Fonte: Elaboração própria (2022).

2.3 Arquiteturas de redes neurais utilizadas em segmentação de imagem

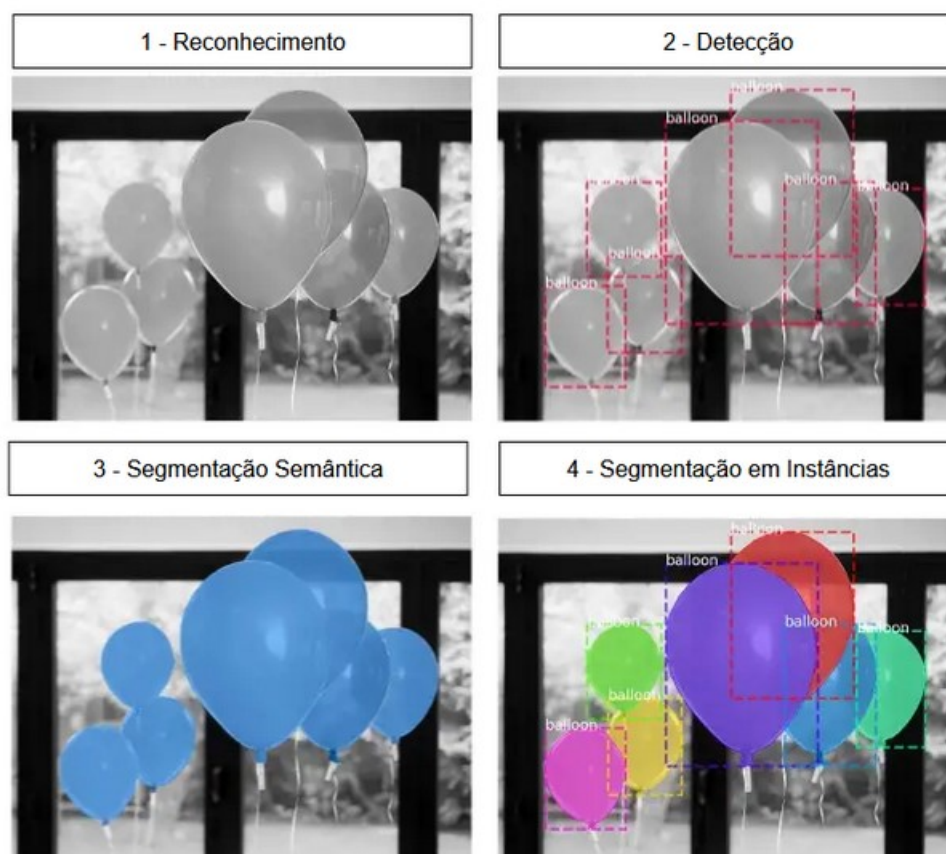
Foi detalhado no capítulo anterior uma das formas de construção de uma rede neural convolucional. O aprendizado profundo utilizado em redes neurais tem uma forte relação sobre a construção de uma arquitetura de rede para corresponder ao problema proposto (PATTERSON; GIBSON, 2017). Com o avanço da utilização das mesmas, além dos avanços tecnológicos que também contribuíram para melhor desempenho, começaram a surgir diferentes combinações de organização e profundidade das camadas para reconhecimento, detecção e/ou segmentação de objetos em imagens.

Na visão computacional existe uma diferenciação sobre reconhecimento, detecção e segmentação semântica de objetos em imagens. Redes que fazem reconhecimento de objetos conseguem determinar se uma imagem contém um determinado objeto. Já redes que determinam onde o objeto se encontra na imagem fazem a detecção de objetos. Caso a rede consiga determinar quais são os pixels referentes ao objeto encontrado na imagem, diz-se que essa rede faz a segmentação semântica. E por fim existem redes que fazem tanto a detecção quanto a segmentação dos objetos em imagens, fazendo então a segmentação em instâncias.

A Figura 13 exibe um exemplo para clarificar a diferença entre as quatro diferentes tarefas com balões. Uma visão computacional que realiza somente o reconhecimento consegue nos informar que existe balão na imagem. Caso ela consiga fazer a detecção de objetos, então a informação que ela disponibilizará será que existem

sete balões na imagem. Já uma rede que faz a segmentação semântica informará quais são os pixels que foram identificados como balão. A segmentação em instâncias é feita por uma rede neural que consegue informar que existem 7 balões, assim como identificar quais pixels são referentes a cada um dos 7 balões.

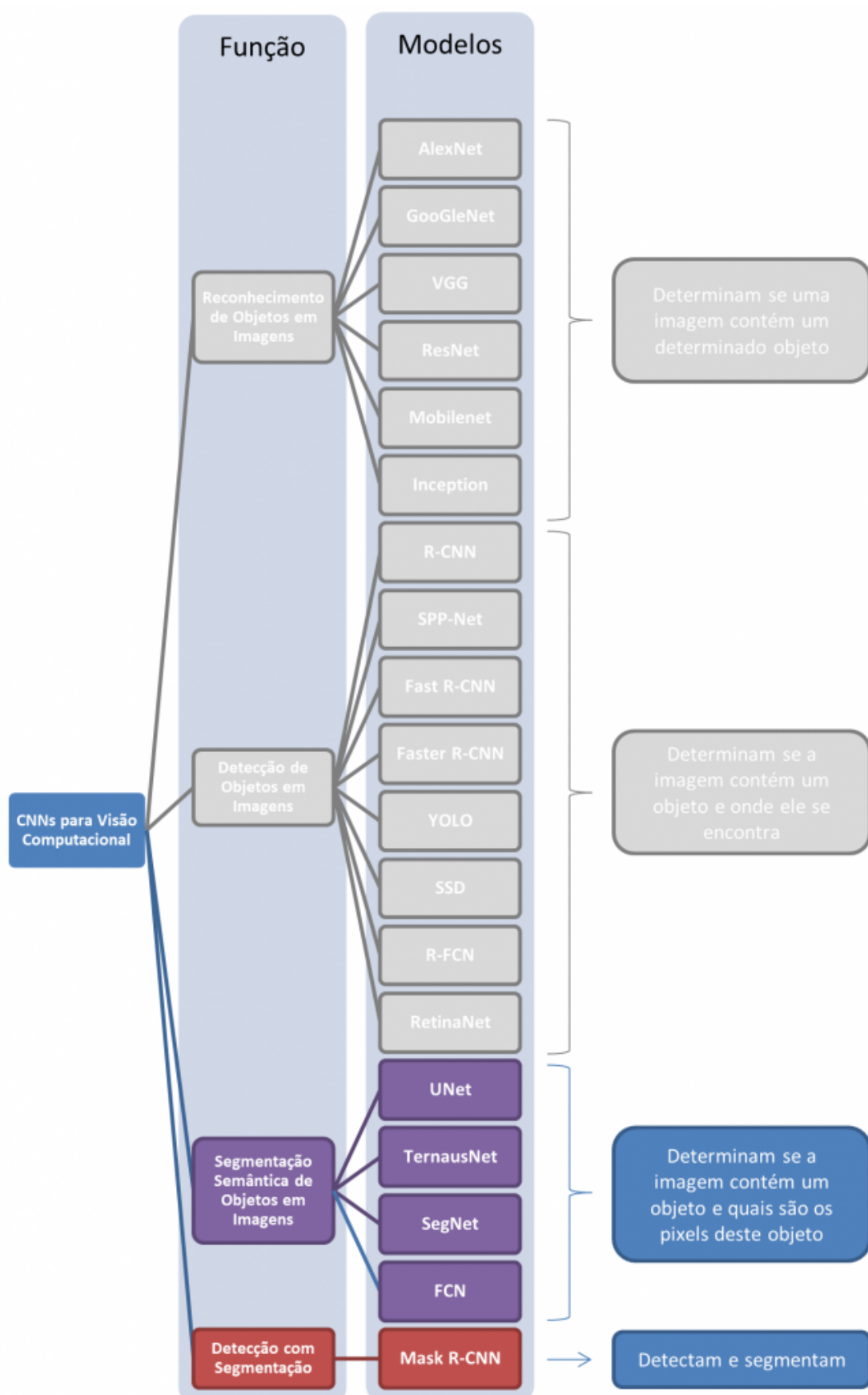
Figura 13 – Exemplo de reconhecimento, detecção e segmentação semântica de objetos.



Fonte: Adaptado de Abdulla (2018).

A Figura 14 exibe um panorama dos modelos de redes neurais convolucionais mais difundidas no campo de visão computacional, além da função de cada uma delas.

Figura 14 – Panorama das CNNs.



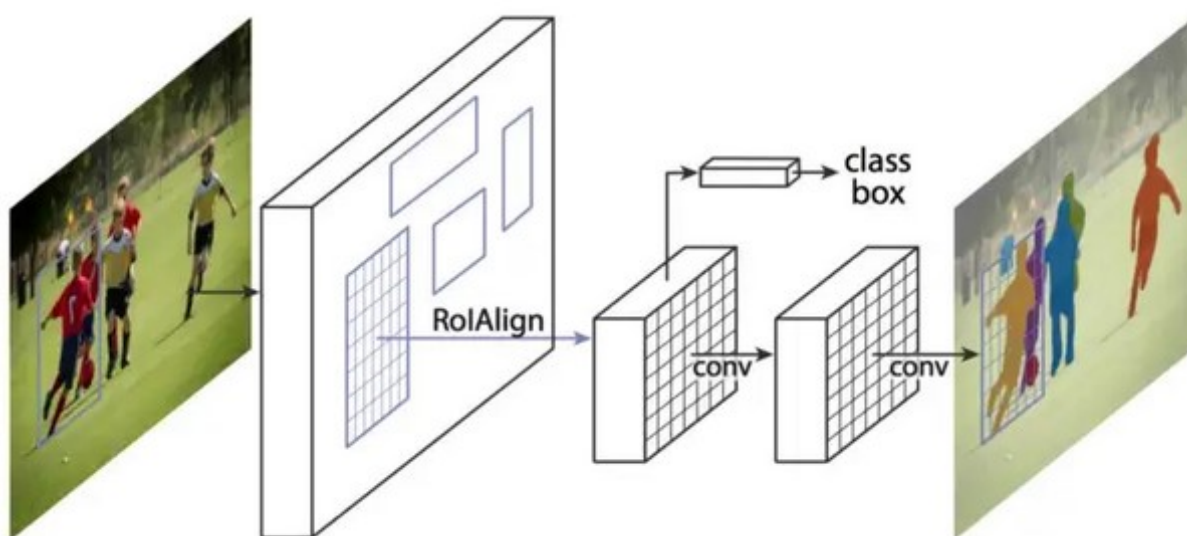
Fonte: (WANGENHEIM, 2018).

Percebe-se a vasta quantidade de arquiteturas e modelos disponíveis para reconhecimento, detecção e segmentação de imagens. Cada modelo possui sua própria estrutura com diferentes convoluções, funções de ativação e pooling. Por ser um trabalho que propõe contagem de laranjas, é necessária a detecção com segmentação. Para isto utilizou-se duas redes baseadas na estrutura Mask R-CNN.

2.3.1 Mask R-CNN

Essa rede neural foi desenvolvida por Kaiming He, Georgia Gkioxari, Pirot Dólar e Ross Girshick, quatro cientistas da FAIR (*Facebook Artificial Intelligence Research*). Eles aprimoraram a anterior rede Faster R-CNN, também criada por eles, adicionando uma ramificação para prever máscaras de segmentação, de onde vem o *mask* no nome. A Figura 15 ilustra a estrutura da Mask R-CNN.

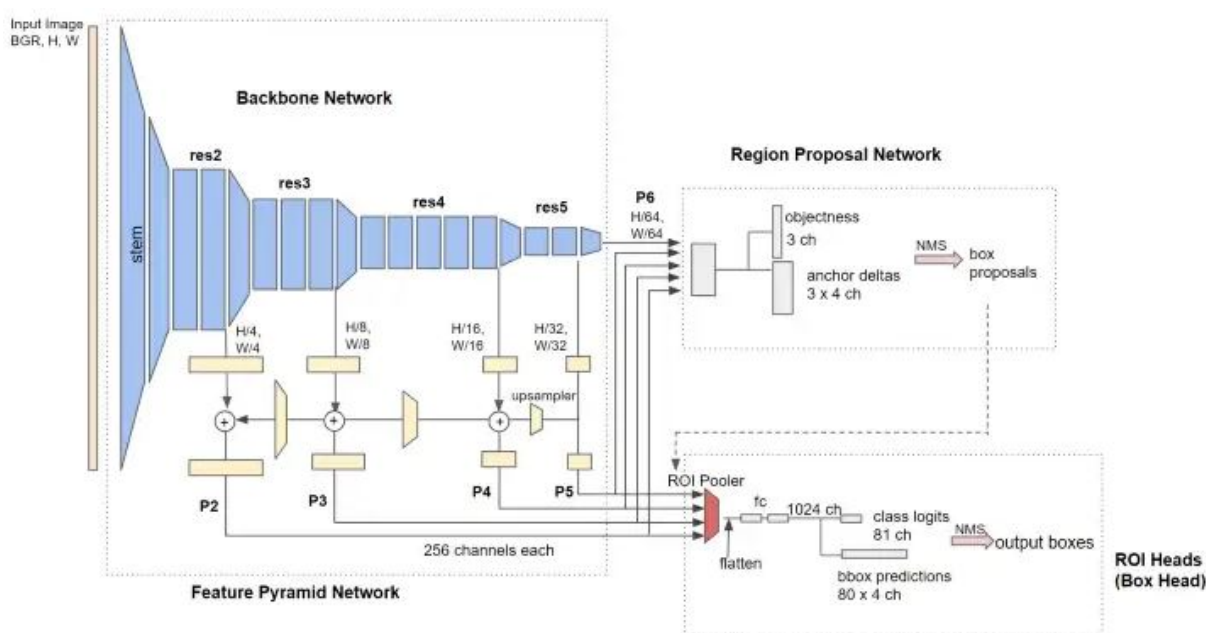
Figura 15 – Estrutura da Mask R-CNN.



Fonte: (HE et al., 2017).

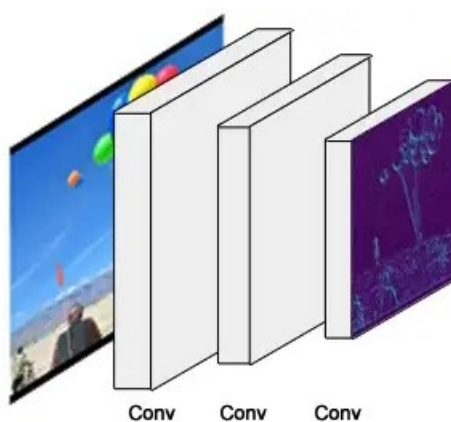
A Mask R-CNN é uma arquitetura estruturada em dois estágios, onde a primeira escaneia a imagem e gera regiões onde há uma possibilidade de conter um objeto, e a segunda faz a classificação dessas regiões gerando caixas delimitadoras e máscaras. De forma resumida, a Mask R-CNN consiste de quatro módulos: (1) *Backbone*; (2) Rede de Propostas Regionais - RPN (*Region Proposal Network*); (3) Classificador de ROI (região de interesse - *region of interest*) e Regressor de Caixa Delimitadora; (4) Máscaras de Segmentação. A Figura 16 exhibe detalhadamente a arquitetura de uma Mask R-CNN, com foco para as conexões entre os módulos mencionados.

Figura 16 – Estrutura detalhada de uma Mask R-CNN.



Fonte: (HONDA, 2020).

O primeiro módulo é conhecido como Backbone, sendo tipicamente implementado por uma rede neural convolucional do tipo ResNet50 ou ResNet101. Ela realiza a extração de informações tanto de baixo nível (cantos e arestas) nas camadas iniciais, quanto às informações de alto nível (pessoa, carro, céu) nas camadas finais, ilustrado na Figura 17. Após esse módulo, a imagem é convertida para um mapa de características que servirá de entrada para os próximos estágios.

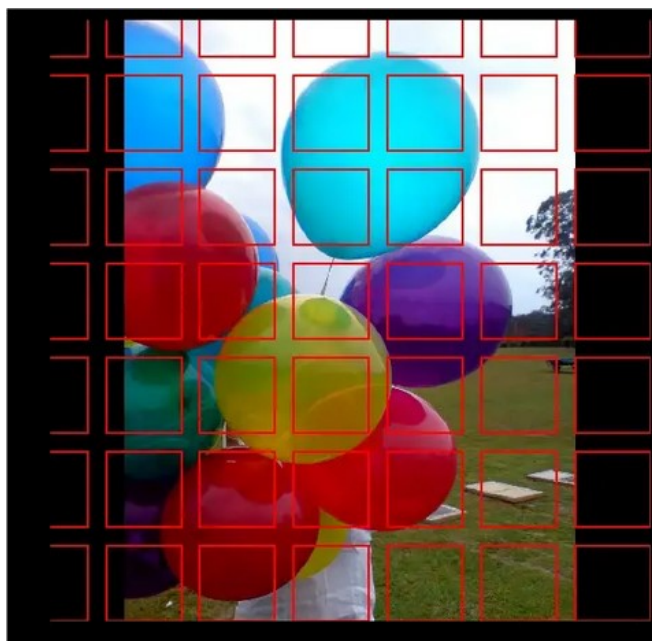
Figura 17 – Rede *Backbone* simplificada.

Fonte: (ABDULLA, 2018).

O segundo módulo, por sua vez, faz o mapeamento da imagem com uma janela deslizante e localiza áreas que contêm objetos. Âncoras é o nome dado para

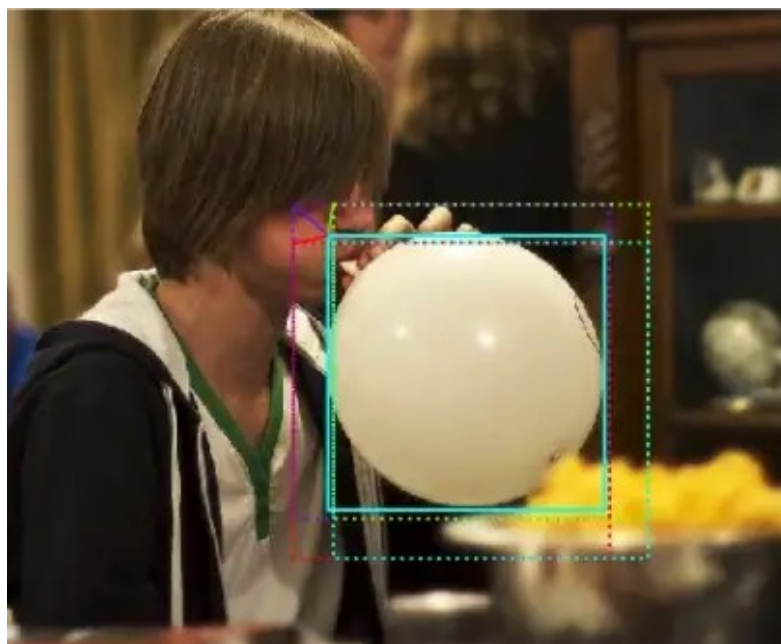
as regiões que a rede RPN sugere, sendo caixas distribuídas pela área da imagem, ilustrado de maneira simplificada na Figura 18. Na prática existem mais de 200 mil âncoras com tamanhos e proporções diferentes, além de se sobreporem, de forma a cobrir o máximo possível da imagem. É importante salientar que a RPN varre o mapa de recursos fornecido pelo backbone, e não a imagem em si, conseguindo reutilizar os recursos extraídos com eficiência e evitando cálculos duplicados.

Figura 18 – Rede RPN exibindo 49 âncoras em uma imagem.



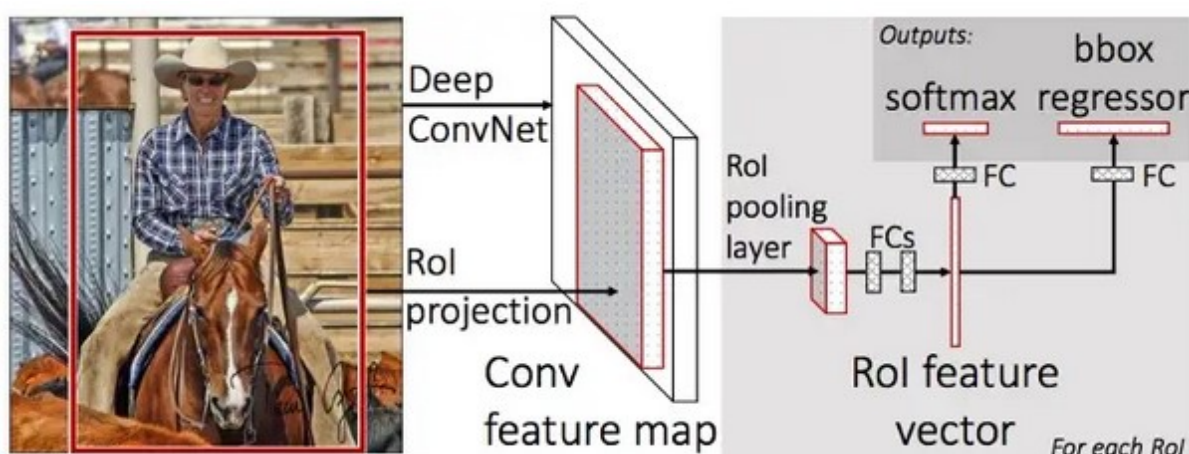
Fonte: (ABDULLA, 2018).

Para cada âncora, a rede RPN gera duas saídas, a classe âncora e a caixa de refinamento. A classe âncora definirá se aqueles pixels representam um primeiro plano ou plano de fundo, onde caso seja o primeiro plano há uma possibilidade de existir um objeto nessa caixa. Já a caixa de refinamento auxilia a centralizar o objeto da classe âncora de primeiro plano. A Figura 19 exibe três âncoras, em pontilhado, e o ajuste aplicado em cada uma delas para enquadrar o objeto de forma precisa, sendo a linha mais grossa. É possível que diferentes âncoras mapeiem o mesmo objeto, todavia são mantidos aqueles com maior pontuação na classe de primeiro plano e descartados o resto. Ao final da RPN tem-se então as regiões de interesse, que serão passadas adiante.

Figura 19 – Regiões de Interesse (RoI) gerados pela rede RPN.

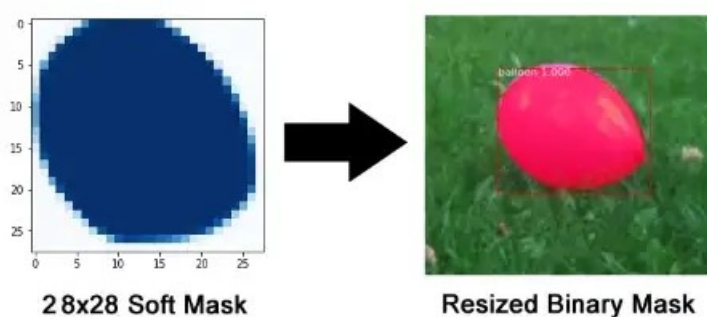
Fonte: (ABDULLA, 2018).

Em seguida tem-se o classificador de região de interesse, que por ser uma rede neural mais profunda tem a capacidade de classificar as regiões de interesse em classes específicas, como por exemplo carro, cadeira, pessoa, etc. Esse classificador também pode descartar a região de interesse caso a mesma seja classificada como uma classe de plano de fundo. O regressor de caixa delimitadora atua de forma similar ao módulo RPN anterior, ajustando a localização e tamanho da delimitação para encapsular o objeto. As etapas e a saída desse terceiro módulo estão exibidas na Figura 20. A camada de agrupamento de região de interesse (RoI pooling layer) auxilia no redimensionamento para um tamanho fixo das regiões de interesse, haja visto que classificadores não conseguem performar bem com tamanhos variados. Os próprios autores do Mask R-CNN indicam utilizar um método de alinhamento de região de interesse, conhecido também como ROIAlign. Nesse método o mapa de recursos é amostrado em diferentes pontos e aplica-se uma interpolação bilinear.

Figura 20 – Classificador de Rol e Regressor de caixa delimitadora.

Fonte: (GIRSHICK, 2015).

Por fim há as máscaras de segmentação, exibido na Figura 21, sendo esse último módulo o que difere uma Faster R-CNN de uma Mask R-CNN. Esse módulo final é uma rede convolucional que, com base nas regiões positivas indicadas no classificador ROI, ou seja, as regiões que existem um objeto, gera máscaras de baixa resolução. No exemplo abaixo tem-se uma máscara soft de 28 x 28 pixels, que diferentemente da máscara binária, é representada em ponto flutuante carregando assim mais detalhes. Dessa forma a previsão da máscara por pixel pode evitar a competição entre diferentes classes.

Figura 21 – Máscara de segmentação gerada na Mask R-CNN.

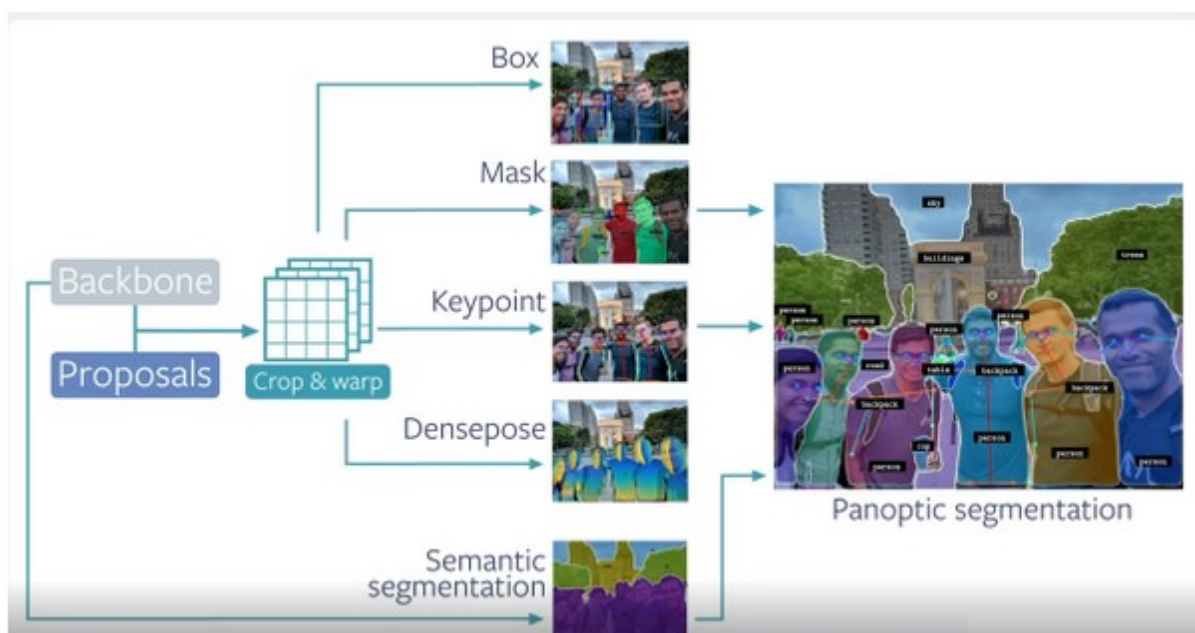
Fonte: (ABDULLA, 2018).

2.3.2 Detectron2

Lançada primeiramente em 2018 como um projeto de código aberto da FAIR (Facebook AI Research), a segunda versão da Detectron foi reescrita para o framework PyTorch. Ela começou com a implementação de uma Mask R-CNN, anteriormente feita

no framework Caffe2. Segundo os Yuxin Wu, et al. o Detectron2 inclui implementações de algoritmos de detecção de objetos de última geração, incluindo DensePose e inúmeras variantes da família pioneira de modelos Mask R-CNN. As Figura 22 e Figura 23 exibem o poder dessa biblioteca, com as possíveis saídas na primeira e uma segmentação panóptica na segunda.

Figura 22 – Diferentes tipos de detecções feitas com Detectron2.



Fonte: Adaptado de Wu et al. (2019).

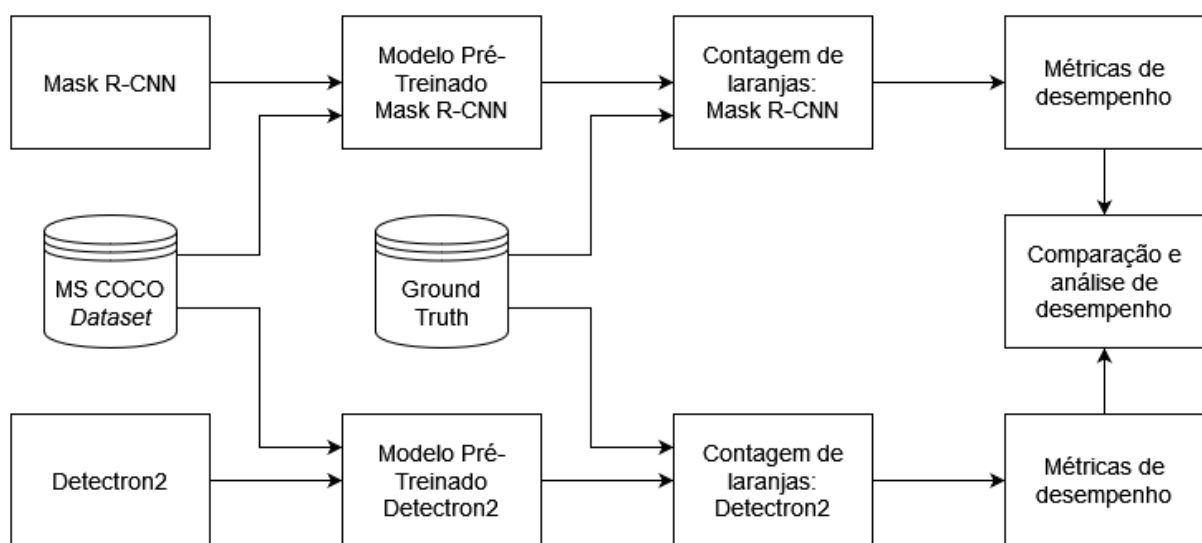
Figura 23 – Segmentação panóptica feita com Detectron2.

Fonte: Adaptado de Wu et al. (2019).

3 DESENVOLVIMENTO

Neste capítulo serão discutidas a escolha das imagens e as etapas de preparação do ground truth. Serão exibidos também as redes neurais e os respectivos algoritmos utilizados, sendo os modelos de Mask R-CNN e Detectron2, ambos já pré treinados com o MS COCO dataset. Por fim, essas redes neurais pré treinadas farão a identificação do número de laranjas do ground truth, onde métricas de desempenho serão levantadas para posterior discussão. O fluxograma do trabalho desenvolvido está exibido na Figura 24

Figura 24 – Diagrama de validação das redes pré-treinadas sob o mesmo *dataset*.



Fonte: Elaboração própria (2022).

É importante destacar que o treino de uma rede neural é uma etapa que demanda tempo, pois esse processo consiste desde a coleta de imagens até a identificação dos pixels que serão classificados, também conhecido como rotulagem, além das métricas de treino. Portanto a escolha dessas duas redes neurais, Mask-RCNN e Detectron2, deve-se a ter encontrado ambos modelos pré-treinados sob a mesma base de dados, MS COCO *Dataset*, acelerando assim o desenvolvimento do trabalho.

3.1 Levantando o *Ground Truth*

Tipicamente, para a implementação de uma rede neural é preciso uma base de dados para treino e outra para teste. Como ambas que iremos utilizar estão treinadas, foi preciso levantar um dataset de teste. O objetivo das redes neurais utilizadas é realizar a contagem de laranjas ainda no pé, logo foram pesquisadas imagens de laranjeiras no

google e contadas de forma manual, com inspeção humana, a quantidade de laranjas em cada uma das imagens.

Ao total foram levantadas 34 imagens de laranjeiras com 33 variações de resolução, que vão de 219 x 183 até 5688 x 3792 pixels. Com essa quantidade diferente de tamanho das imagens tem-se uma base de teste heterogênea. Para a contagem da quantidade de laranjas por inspeção humana em cada imagem foi considerado qualquer tamanho e profundidade de imagem que continha a fruta.

As imagens utilizadas, assim como a contagem realizada manualmente, estão presentes em Netto e Bonho (2021). As Figura 25, Figura 26 e Figura 27 a seguir exibem exemplos das imagens para teste do algoritmo.

Figura 25 – Laranjeira do *ground truth*, contagem manual de 42 laranjas.



Fonte: Elaboração própria (2022).

Figura 26 – Laranjeira do *ground truth*, contagem manual de 68 laranjas.



Fonte: Elaboração própria (2022).

Figura 27 – Laranjeira do *ground truth*, contagem manual de 16 laranjas.



Fonte: Elaboração própria (2022).

3.2 Métricas utilizadas para comparação das redes neurais

Foram utilizadas três métricas para comparação das redes neurais, sendo média da acurácia individual, variância e coeficiente de determinação. A acurácia

individual indicará quantas laranjas a rede neural conseguiu encontrar em comparação ao *ground truth*, conforme exige a Equação 3.1.

$$\text{Acurácia Individual} = \frac{\text{Qtd laranjas detectadas pelo modelo}}{\text{Qtd laranjas contadas no ground truth}} \quad (3.1)$$

A média da acurácia individual, por sua vez, é a divisão do somatório das acurácias individuais pela quantidade de imagens, Equação 3.2. Essa métrica indica quão bom foi o desempenho do modelo para detectar as laranjas do *ground truth*.

$$\overline{\text{Acurácia Individual}} = \frac{\sum \text{Acurácia Individual}_n}{\text{Qtd. Imagens Ground Truth}} \quad (3.2)$$

Na estatística a variância amostral indica o comportamento de uma amostra em relação a uma medida central, indicando o quão próximo os valores estão em relação a média. A variância populacional, por sua vez, indica o comportamento de todas as amostras. De forma semelhante, no presente trabalho, utilizou-se a variância populacional para indicar a dispersão das laranjas detectadas pelo modelo em relação a média das laranjas do *ground truth*. A Equação 3.3, Equação 3.4 e Equação 3.5 detalham como foi feito o cálculo da variância populacional.

$$x_n = \text{Número de laranjas da amostra } n \quad (3.3)$$

$$\bar{x} = \frac{\text{Qtd. laranjas no ground truth}}{\text{Qtd. imagens do ground truth}} \quad (3.4)$$

$$\sigma^2 = \frac{(x_1 - \bar{x}) + \dots + (x_n - \bar{x})}{n} \quad (3.5)$$

Por fim utilizou-se o coeficiente de correlação de Pearson, também conhecido como ρ , que mede o grau da correlação entre duas variáveis. O ρ varia entre -1 e 1, indicando uma correlação negativa ou positiva. No presente trabalho o ρ indica o quão próximo o modelo consegue explicar o comportamento dessa problemática. Portanto, quanto mais próximo de 1, melhor será o desempenho do modelo na contagem de laranjas do *ground truth*. A Equação 3.6 detalha o cálculo.

Vale ressaltar que $x_1, x_2, \dots, x_n$ são os valores referentes a contagem de laranjas no *ground truth*, $y_1, y_2, \dots, y_n$ são os valores referentes a contagem de laranjas pelo modelo. Além disso, a média de cada uma dessas variáveis está na Equação 3.7 e Equação 3.8.

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}} \quad (3.6)$$

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \quad (3.7)$$

$$\bar{y} = \frac{1}{n} \cdot \sum_{i=1}^n y_i \quad (3.8)$$

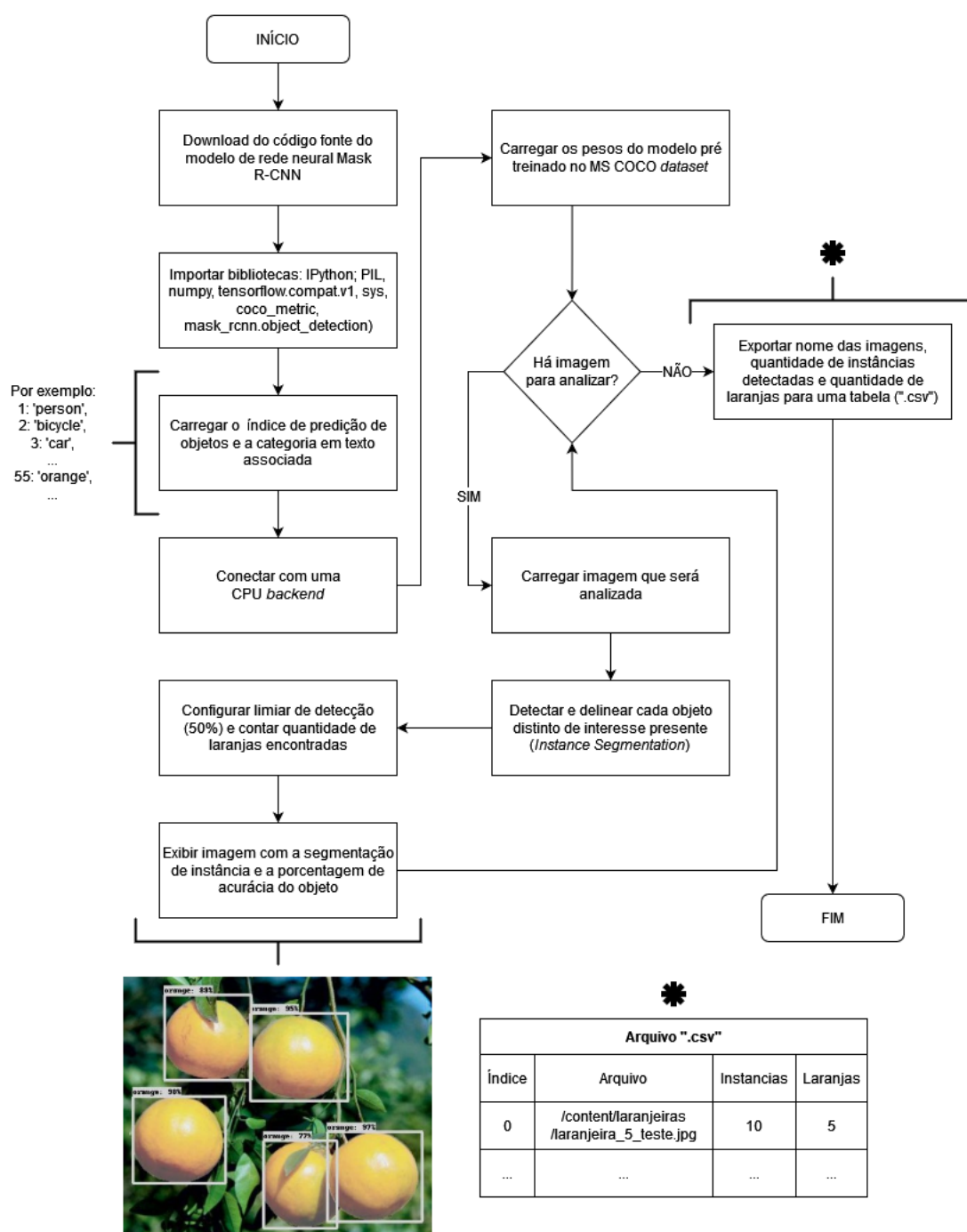
3.3 Algoritmo utilizado para rede neural Mask R-CNN

Na documentação da rede neural Mask R-CNN existem diferentes exemplos para treinar e testar a rede neural, dentre eles há exemplos de identificação de objetos utilizando os pesos de uma rede neural pré-treinada. O algoritmo a seguir, disponível em Silva e Bonho (2022b) foi desenvolvido baseado em um exemplo, sendo testado a identificação de laranjas ainda no pé.

Um ponto importante a se destacar nesse script é o mapeamento entre índice de predição de objetos e a categoria em texto associado a ele. Isso é necessário pois foi utilizado uma rede neural generalista, ou seja, ela foi treinada para identificar cerca de 90 objetos diferentes com a base de treino MS COCO. Por exemplo, o índice “1” corresponde a categoria “pessoa”, o “2” a “bicicleta”, “3” o “carro”, e assim sucessivamente, sendo a “laranja” o índice “55”.

Logo, como o foco é identificar laranjas, utilizou-se o índice “55” para filtrar o dado de interesse e exportar o número de detecções em uma planilha. A Figura 28 a seguir exibe o fluxograma do algoritmo desenvolvido.

Figura 28 – Fluxograma do algoritmo desenvolvido aplicando a rede neural Mask R-CNN.



Fonte: Elaboração própria (2022).

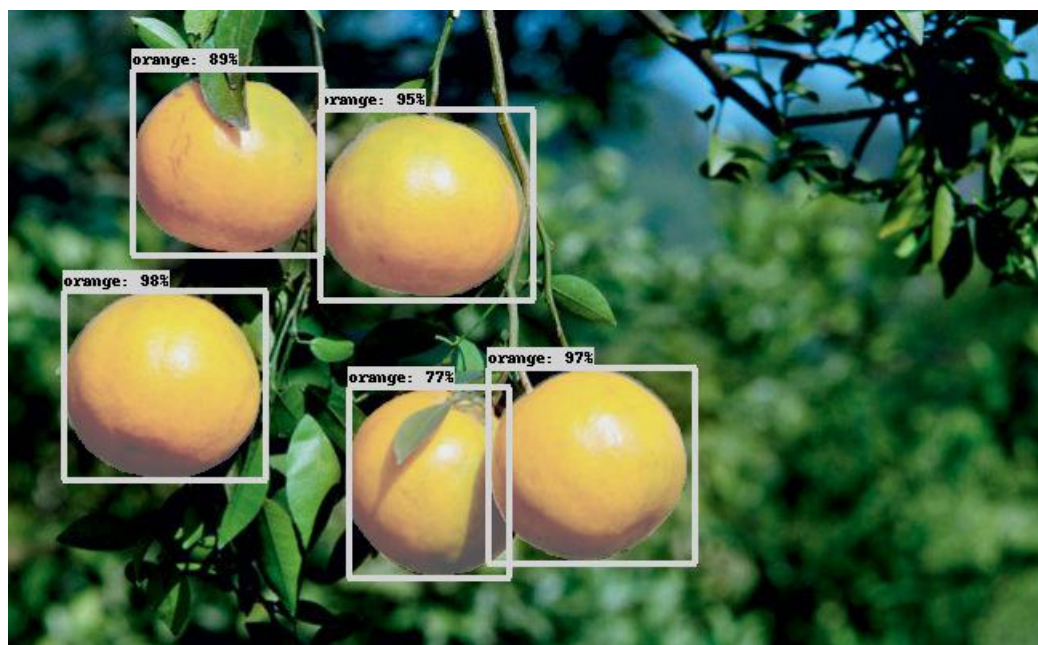
Para testar o algoritmo, utilizou-se uma imagem de fácil inspeção visual, contendo 5 laranjas, conforme exibe a Figura 29. O resultado do algoritmo desenvolvido é ilustrado na Figura 30.

Figura 29 – Laranjeira utilizada para teste do algoritmo com inspeção visual de 5 laranjas.



Fonte: Elaboração própria (2022).

Figura 30 – Laranjeira após algoritmo de segmentação de imagens com Mask R-CNN.



Fonte: Elaboração própria (2022).

Observou-se que o algoritmo conseguiu identificar, demarcar e contabilizar as 5 laranjas na imagem proposta. Além disso, notou-se que a porcentagem de precisão na detecção dessas laranjas variou entre 77% e 98%. Logo, dependendo de como

a foto foi capturada, o algoritmo pode ter mais facilidade ou dificuldade de fazer a identificação do objeto em questão.

3.4 Aplicando a rede neural Mask R-CNN pré-treinada com MS COCO *dataset* no *Ground Truth*

Aplicou-se então o algoritmo desenvolvido na base de teste, comparando o desempenho da contagem de laranjas da rede neural em cada imagem, tendo assim uma acurácia individual. A Tabela 2 detalha a quantidade de instâncias, ou seja, classes detectadas, o número de laranjas contabilizado pelo algoritmo, a quantidade de laranjas no *ground truth* e a acurácia individual.

Tabela 2 – Contagem de laranja da rede neural Mask R-CNN e acurácia individual.

Nome da figura	Instancias	Laranjas	Ground Truth	Acurácia Individual
laranjeira_1.jpg	66	28	35	80,00%
laranjeira_10.jpg	63	20	34	58,82%
laranjeira_11.jpg	51	26	39	66,67%
laranjeira_12.jpg	100	59	84	70,24%
laranjeira_13.jpg	86	22	103	21,36%
laranjeira_14.jpg	100	59	84	70,24%
laranjeira_15.jpg	54	11	16	68,75%
laranjeira_16.jpg	80	42	57	73,68%
laranjeira_17.jpg	100	51	101	50,50%
laranjeira_18.jpg	59	19	52	36,54%
laranjeira_19.jpg	95	35	111	31,53%
laranjeira_2.jpg	89	32	42	76,19%
laranjeira_20.jpg	27	9	16	56,25%
laranjeira_21.jpg	45	25	32	78,13%
laranjeira_22.jpg	46	18	29	62,07%
laranjeira_23.jpg	67	25	52	48,08%
laranjeira_24.jpg	62	16	56	28,57%
laranjeira_25.jpg	36	10	21	47,62%
laranjeira_26.jpg	14	5	13	38,46%
laranjeira_27.jpg	45	15	29	51,72%
laranjeira_28.jpg	26	11	17	64,71%
laranjeira_29.jpg	50	12	20	60,00%
laranjeira_3.jpg	100	74	153	48,37%
laranjeira_30.jpg	88	37	68	54,41%
laranjeira_31.jpg	55	20	28	71,43%
laranjeira_32.jpg	72	27	42	64,29%
laranjeira_33.jpg	100	71	105	67,62%
laranjeira_34.jpg	100	39	107	36,45%
laranjeira_4.jpg	68	28	44	63,64%
laranjeira_5.jpg	59	22	36	61,11%
laranjeira_6.jpg	70	31	68	45,59%
laranjeira_7.jpg	34	12	24	50,00%
laranjeira_8.jpg	84	33	61	54,10%
laranjeira_9.jpg	100	36	86	41,86%

Fonte: Elaboração própria (2022).

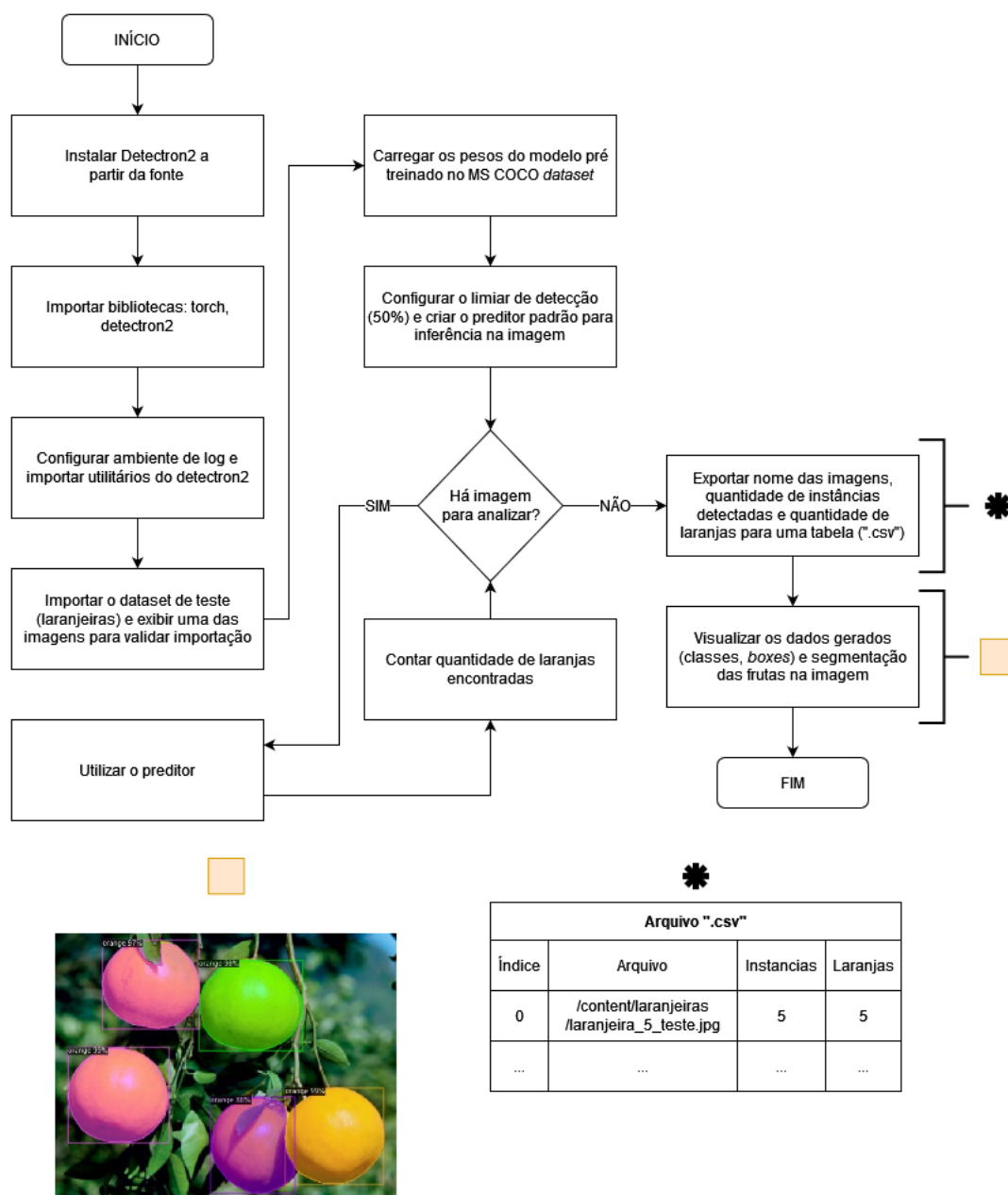
Observou-se uma variação de acurácia individual de 21,36% até 80,00% . A média da acurácia individual foi de 55,85% e a variância (σ^2) de 0,02159. O coeficiente de correlação (ρ), por sua vez, foi de 0,6924, indicando uma correlação moderada entre o modelo com Mask R-CNN e a base de teste.

3.5 Algoritmo utilizado para rede neural Detectron2

De forma similar, a rede neural Detectron2 também possui uma documentação vasta e scripts exemplos para testar a segmentação de imagens. Foi utilizado o tutorial na documentação como base para criar o algoritmo de teste deste trabalho. Utilizou-se a mesma base de dados de treino, sendo MS COCO dataset, para implementação da rede neural. O código desenvolvido está disponível em Silva e Bonho (2022a).

É importante salientar que apesar das redes neurais terem sido treinadas pela mesma base de dados, o índice da classe laranja no Detectron2 é “49”. Logo utilizou-se o mesmo para filtrar a classe de interesse e realizar a contagem da fruta. A imagem a seguir, Figura 31 exibe o fluxograma do algoritmo desenvolvido.

Figura 31 – Fluxograma do algoritmo desenvolvido aplicando a rede neural Detectron2



Fonte: Elaboração própria (2022).

Utilizando a mesma imagem da rede neural anterior, Figura 29, testou-se o algoritmo para detectar laranjas em uma laranjeira, conforme ilustrado na Figura 32.

Notou-se que o algoritmo conseguiu identificar, demarcar e contabilizar as 5 laranjas na imagem proposta. Além disso, notou-se que a porcentagem de precisão na detecção dessas laranjas variou entre 88% e 99%. O algoritmo em questão fez a segmentação das laranjas em cores diferentes, facilitando a identificação de laranjas sobrepostas.

Figura 32 – Laranjeira após algoritmo de segmentação de imagens com Detectron2.



Fonte: Elaboração própria (2022).

3.6 Aplicando a rede neural Detectron2 pré-treinada com MS COCO dataset no *Ground Truth*

Foi aplicado o script desenvolvido com as imagens do Ground Truth e calculou-se a acurácia individual na contagem de laranjas. A Tabela 3 exibe em detalhes o número de laranjas contadas pela rede neural Detectron2 e no Ground Truth, assim como a acurácia individual calculada.

Tabela 3 – Contagem de laranja da rede neural utilizando Detectron2 e acurácia individual

Nome da figura	Instancias	Laranjas	Ground Truth	Acurácia Individual
laranjeira_1.jpg	34	32	35	91,43%
laranjeira_10.jpg	25	22	34	64,71%
laranjeira_11.jpg	28	28	39	71,79%
laranjeira_12.jpg	50	50	84	59,52%
laranjeira_13.jpg	18	18	103	17,48%
laranjeira_14.jpg	51	51	84	60,71%
laranjeira_15.jpg	13	13	16	81,25%
laranjeira_16.jpg	43	43	57	75,44%
laranjeira_17.jpg	54	54	101	53,47%
laranjeira_18.jpg	27	27	52	51,92%
laranjeira_19.jpg	33	32	111	28,83%
laranjeira_2.jpg	35	33	42	78,57%
laranjeira_20.jpg	10	10	16	62,50%
laranjeira_21.jpg	23	22	32	68,75%
laranjeira_22.jpg	19	19	29	65,52%
laranjeira_23.jpg	29	27	52	51,92%
laranjeira_24.jpg	24	22	56	39,29%
laranjeira_25.jpg	13	12	21	57,14%
laranjeira_26.jpg	7	6	13	46,15%
laranjeira_27.jpg	14	14	29	48,28%
laranjeira_28.jpg	11	10	17	58,82%
laranjeira_29.jpg	16	10	20	50,00%
laranjeira_3.jpg	77	77	153	50,33%
laranjeira_30.jpg	33	33	68	48,53%
laranjeira_31.jpg	18	16	28	57,14%
laranjeira_32.jpg	25	22	42	52,38%
laranjeira_33.jpg	77	77	105	73,33%
laranjeira_34.jpg	49	49	107	45,79%
laranjeira_4.jpg	33	26	44	59,09%
laranjeira_5.jpg	19	19	36	52,78%
laranjeira_6.jpg	31	31	68	45,59%
laranjeira_7.jpg	14	13	24	54,17%
laranjeira_8.jpg	32	30	61	49,18%
laranjeira_9.jpg	33	29	86	33,72%

Fonte: Elaboração própria (2022).

A acurácia individual variou de 17,48% até 91,43%, com média de 56,04% e variância (σ^2) de 0,02165. Já o coeficiente de correlação (ρ) foi de 0,6883, indicando também uma correlação moderada entre o modelo com Detectron2 e a base de teste.

4 DISCUSSÃO

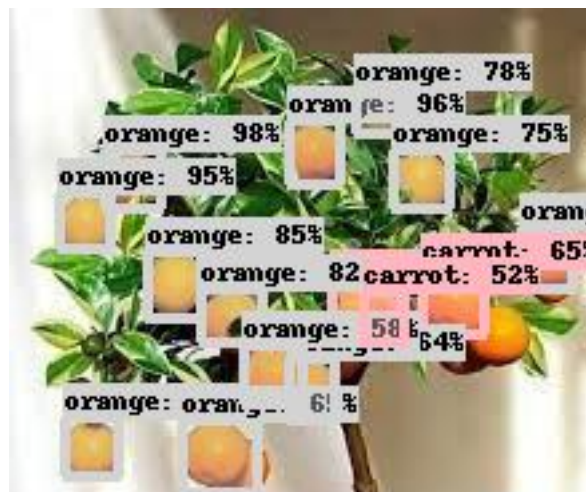
A base de teste levantada possui 33 resoluções diferentes, com isso analisou-se a relação entre resolução e acurácia. Na Figura 33, onde a resolução é de 219 x 183 pixels, totalizando 40077 pixels, a acurácia individual foi de 60,0% na rede neural Mask R-CNN (Figura 34) e de 50,0% na Detectron2 (Figura 35). Por outro lado, a Figura 36, com resolução de 2560 x 1707 pixels, logo 4369920 pixels ao total, a acurácia individual foi de 34,54% (Figura 37) e 51,92% (Figura 38) respectivamente em cada rede neural.

Figura 33 – Imagem com resolução de 219 x 183 pixels.

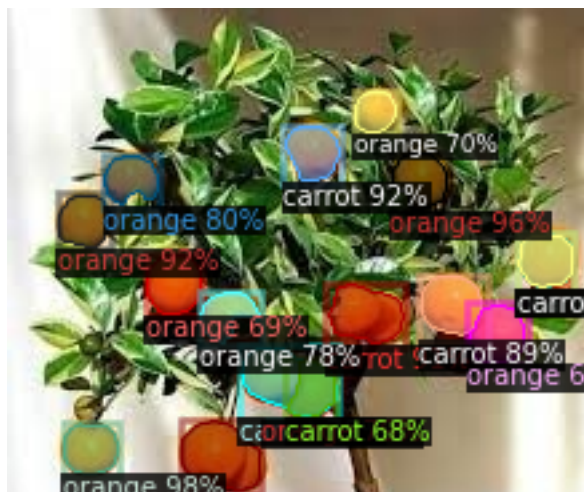


Fonte: Elaboração própria (2022).

Figura 34 – Segmentação com Mark R-CNN.



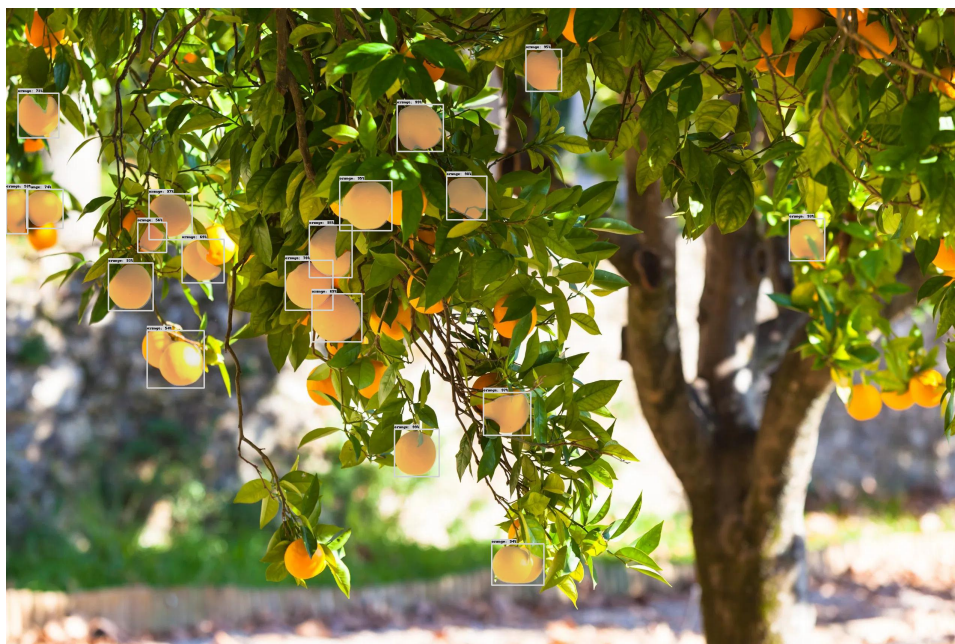
Fonte: Elaboração própria (2022).

Figura 35 – Segmentação com Detectron2.

Fonte: Elaboração própria (2022).

Figura 36 – Imagem com resolução de 2560 x 1707 pixels.

Fonte: Elaboração própria (2022).

Figura 37 – Segmentação com Mask R-CNN.

Fonte: Elaboração própria (2022).

Figura 38 – Segmentação com Detectron2.

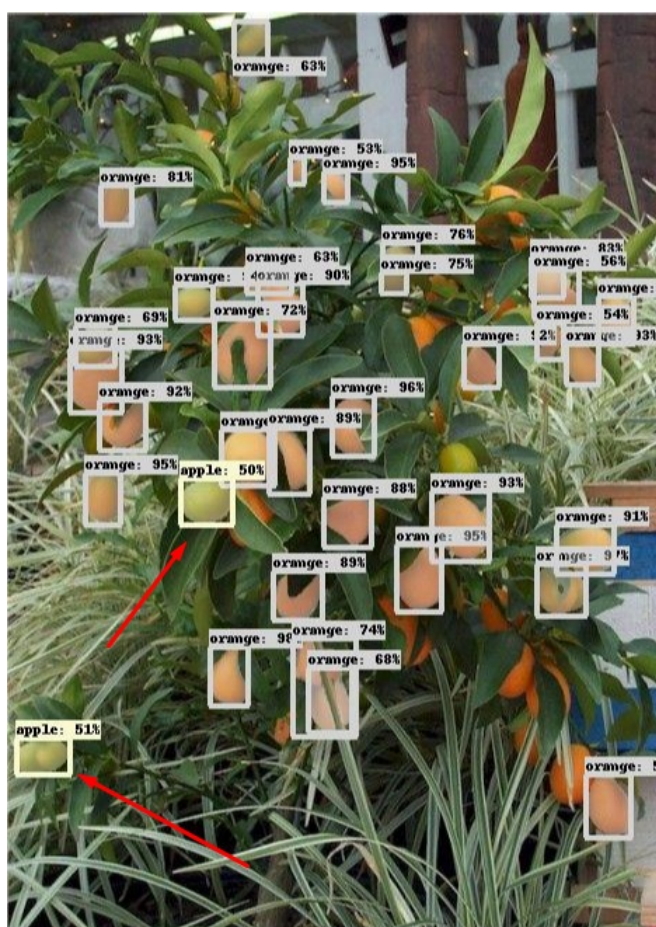
Fonte: Elaboração própria (2022).

Nesse exemplo a resolução da maior imagem é aproximadamente 100 vezes maior. Todavia as acurácias individuais foram inferiores ou muito próximas à acurácia obtida na imagem com menor resolução. Foi possível notar também, em ambos casos,

a presença de outras detecções como cenoura (carrot), além de laranjas diferentes, no caso duas, sendo contadas somente como uma.

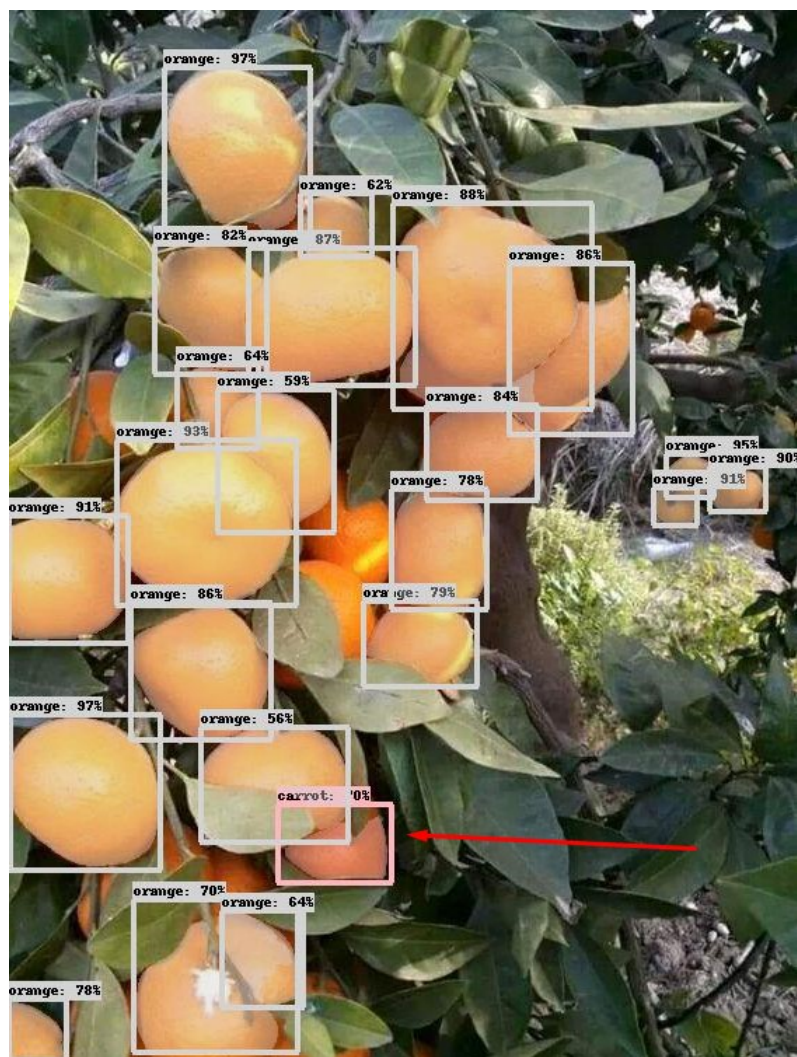
Era de se esperar a detecção de outros objetos, dada a característica generalista das redes neurais convolucionais utilizadas. Todavia, a identificação de uma cenoura onde claramente não existe a possibilidade de existir uma indicam uma problemática de reutilizar redes generalistas em tarefas específicas, como a contagem de laranjas. As Figura 39, Figura 40 e Figura 41 exibem mais segmentações erradas. Poderia ter sido evitado a detecção dessas classes, todavia impactaria na contagem de laranjas, pois tem-se *scores* que variam desde 51% a 98%.

Figura 39 – Segmentação de uma maçã onde não há.



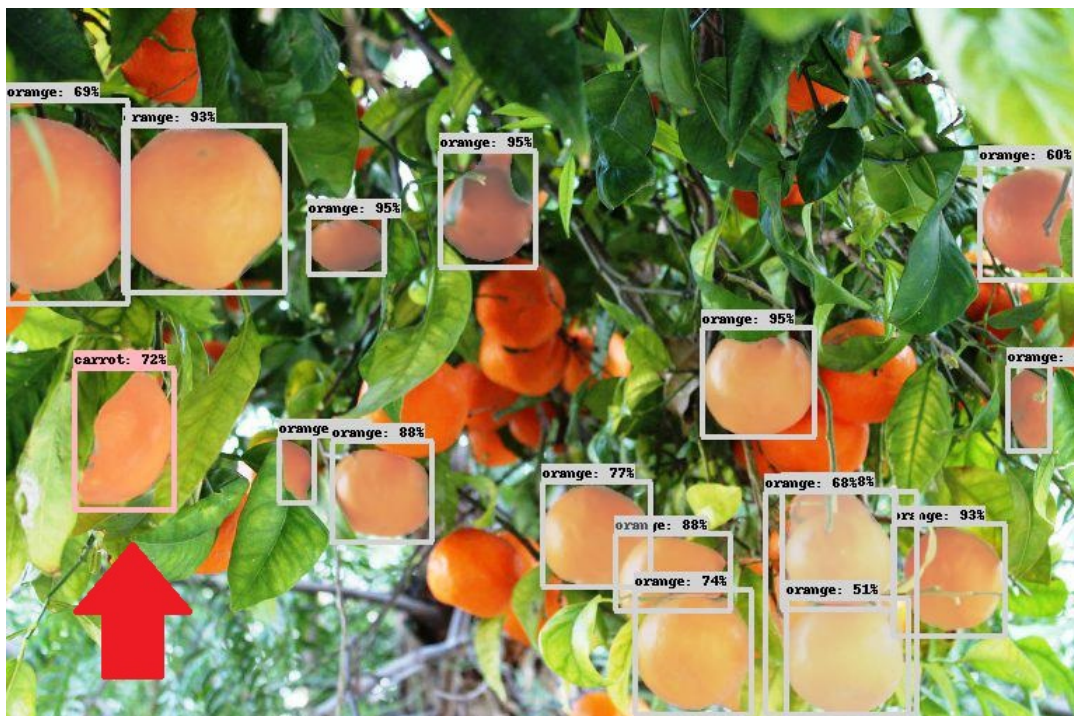
Fonte: Elaboração própria (2022).

Figura 40 – Segmentação de uma cenoura onde não há



Fonte: Elaboração própria (2022).

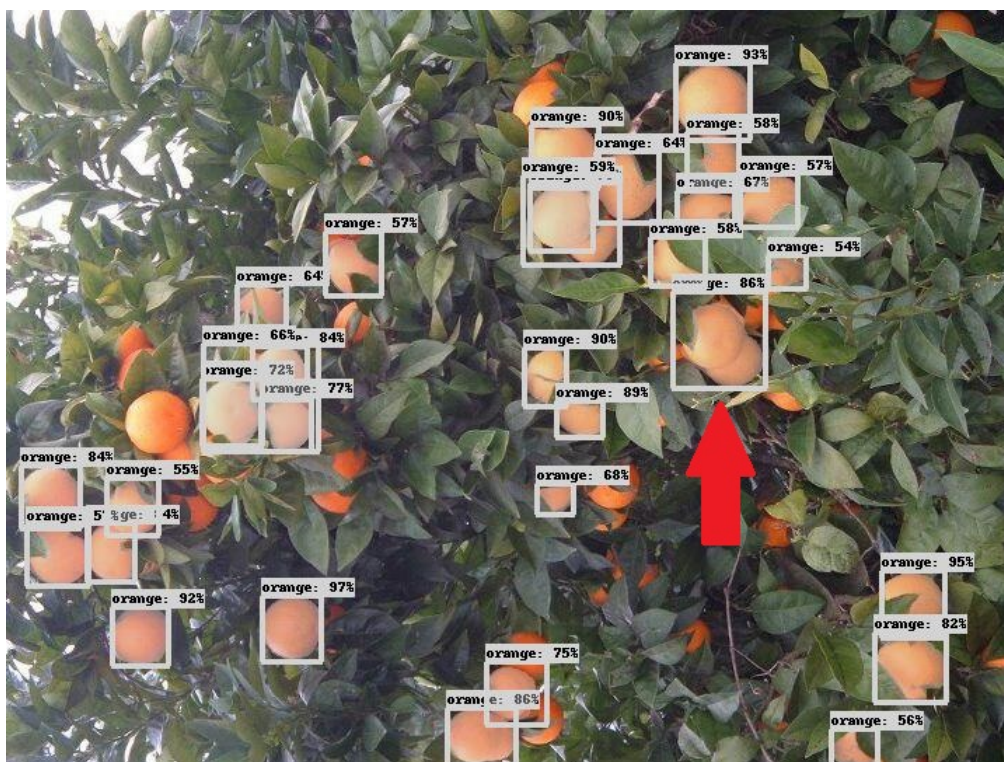
Figura 41 – Segmentação com Detectron2.



Fonte: Elaboração própria (2022).

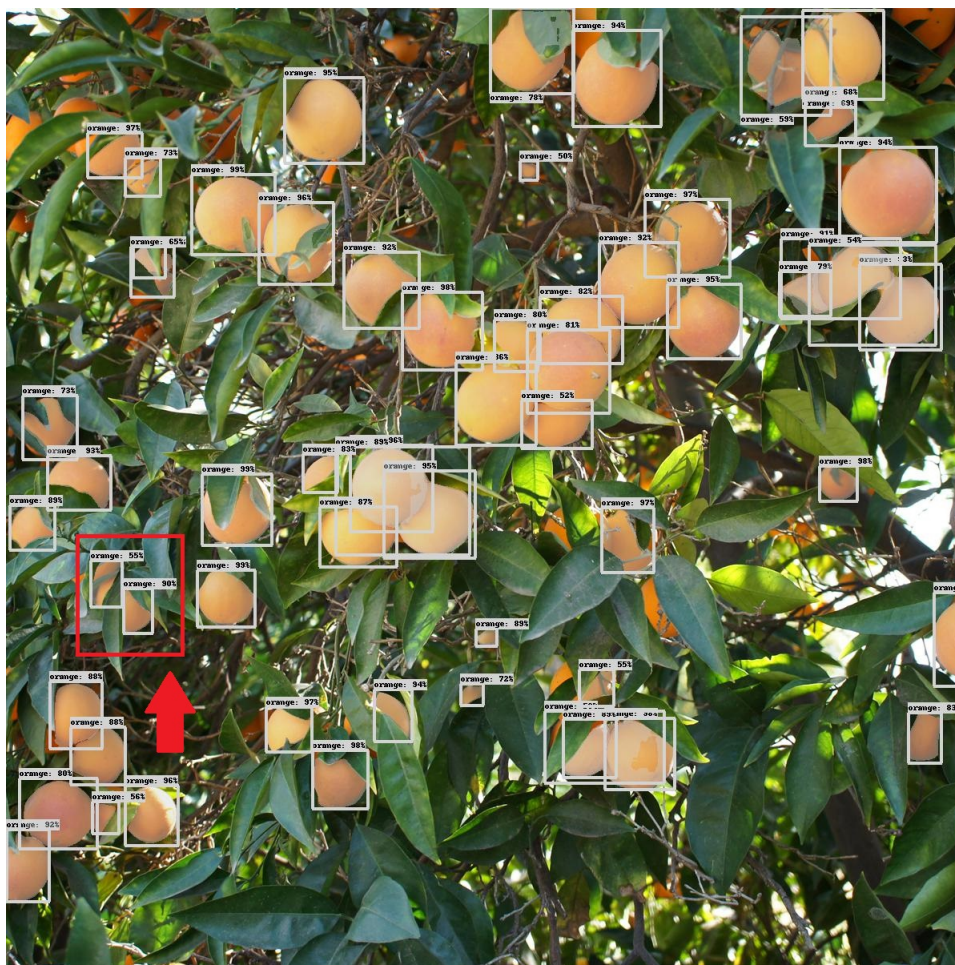
Observou-se também que a rede Mask R-CNN não conseguiu performar bem quando ocorreu a sobreposição de pixels de laranjas, como exibe a Figura 42. A contagem também foi prejudicada com a contagem de uma laranja sendo detectada como duas laranjas pela sobreposição de folhas e galhos, ilustrado nas Figura 43 e Figura 44

Figura 42 – Segmentação de duas laranjas distintas como sendo somente uma.



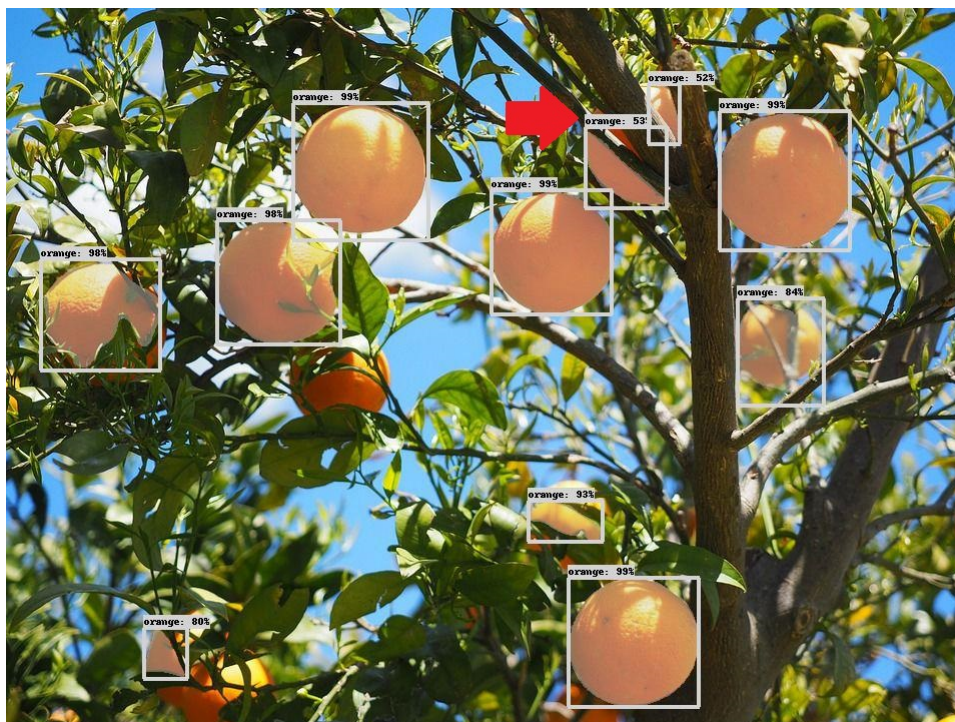
Fonte: Elaboração própria (2022).

Figura 43 – Segmentação de uma laranja como duas pela sobreposição de folhas.



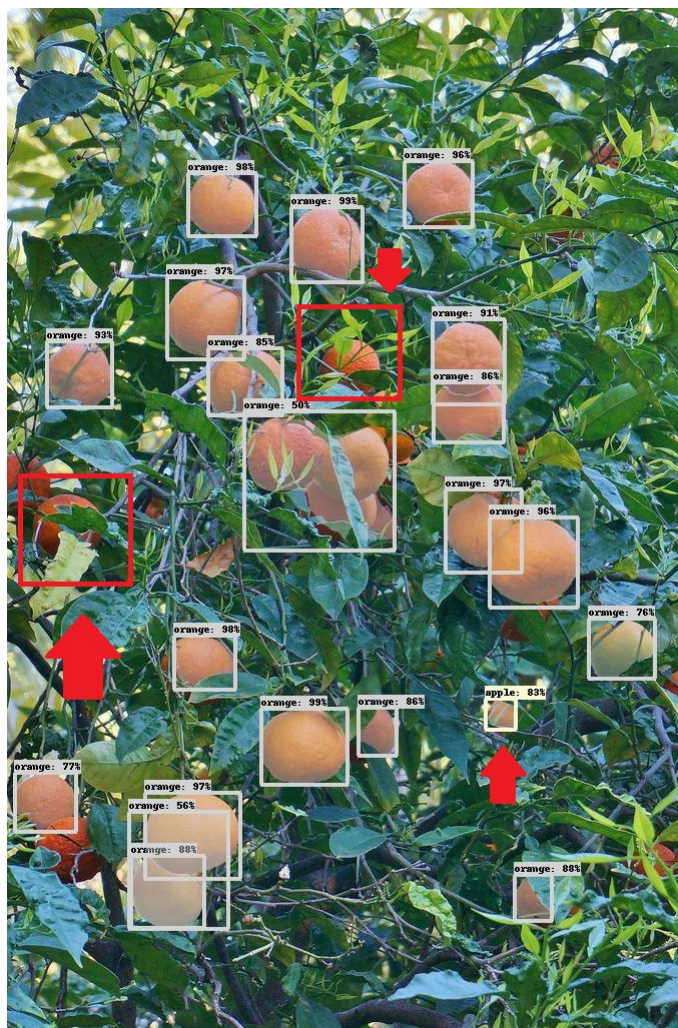
Fonte: Elaboração própria (2022).

Figura 44 – Segmentação de uma laranja como duas pela sobreposição da árvore.



Fonte: Elaboração própria (2022).

Por fim foi analisado também a dificuldade de reconhecer laranjas pela sobreposição de folhas, mostrado na Figura 45, sendo que nesse caso essas laranjas não foram contabilizadas.

Figura 45 – Dificuldade de segmentação pela sobreposição da folhas.

Fonte: Elaboração própria (2022).

Ambas redes apresentaram baixa performance, haja visto que outros trabalhos com as mesmas arquiteturas conseguem desempenhos superiores à 90% de detecção e segmentação de objetos. É importante salientar também os desafios não só na segmentação da imagem, mas também na aquisição da mesma, pois como foi visto a sobreposição de ruídos como folhas e galhos pode interferir de forma significativa na contabilização das laranjas.

5 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou um estudo comparativo entre duas redes neurais convolucionais para detecção de laranjas a partir da segmentação de imagens de laranjeiras, cada imagem com diferentes resoluções. A ideia principal foi buscar duas arquiteturas de redes bastante utilizadas e conhecidas, para testá-las em um dataset selecionado.

Dentre as arquiteturas conhecidas, selecionaram-se as redes Mask R-CNN e Detectron2. Ambas estruturas já disponibilizavam toda rede treinada e seus respectivos pesos das camadas neurais escondidas. Para o treino, as duas arquiteturas utilizaram o MS COCO, cujo dataset é mundialmente conhecido e utilizado em tarefas de detecção de objetos e segmentação. Por ter mais de 330 mil imagens (cerca de 200 mil com o label já definido para o caso de aprendizagem supervisionada) e 80 categorias de objeto, a grande maioria das novas propostas de redes convolucionais usadas em visão computacional fazem uso deste banco de dados.

Buscou-se saber o quanto genérica estas estruturas são em relação a um problema muito específico de detecção de objeto, no caso laranjas. O dataset para teste utilizado é composto por 34 imagens de laranjeiras com 33 variações de resolução, de 219 x 183 até 5688 x 3792 pixels. O ground truth de cada imagem foi obtido através de inspeção visual humana.

No processo de segmentação de imagem com a rede Mask R-CNN, obteve-se uma acurácia média de 55,85%, uma variância (σ^2) de 0,02159 e um ρ de 0,6924. Utilizando a arquitetura da rede Detectron2, a acurácia média atingida foi de 56,04%, variância (σ^2) de 0,02165 e ρ de 0,6883.

Na área de segmentação de imagem, o aumento da resolução da imagem traz como consequência um acréscimo do número de segmentos e objetos reconhecidos. Em classificação de imagem de radiografia, satélites e MRI, por exemplo, sistemas que utilizam um número maior de pixels para a mesma área (maior resolução espacial) tendem a ter uma melhor performance.

Entretanto, a acurácia na classificação das laranjas, obtida pelas duas redes, não teve uma relação linear com a qualidade das imagens. Acredita-se que o fato das redes terem sido treinadas para segmentar e classificar 80 categorias de objetos, não faz com que a acurácia seja maior para maiores resoluções. Soma-se a isto o fato das imagens das laranjas, além de possuírem profundidade, terem sobreposição com imagens das folhas da laranjeiras. Quando a rede neural utiliza regiões de interesse (ROI), a segmentação e reconhecimento dos pixels que compõem a imagem podem formar um padrão que não foi mapeado no treinamento, o que explicaria também a acurácia moderada na classificação.

É importante ressaltar também a importância de se treinar a rede com imagens similares às que serão classificadas. Por exemplo, não adiantaria treinar uma rede neural com fotos de laranjas em cima da mesa, logo em um ambiente com pouco ruído. Esse modelo certamente não teria um desempenho bom ao tentar detectar e contabilizar laranjas em uma laranjeira, logo em um ambiente com ruído de folhas e galhos. Os padrões nos dois cenários são diferentes, o que influenciará nas regiões de interesse que a rede neural irá gerar.

Como trabalhos futuros sugere-se treinar as arquiteturas com um dataset formado por somente imagens de laranjeiras, com a máscara de cada imagem informando as áreas onde se encontram os frutos. Há também a possibilidade de mensurar o tempo de classificação de cada arquitetura, para um estudo de viabilidade de aplicação da rede neural em sistemas embarcados, usando o conceito de computação de borda (*edge computing*). Geralmente, redes muito complexas são um empecilho para o processamento nos próprios dispositivos utilizados pelos usuários e podem ter uma baixa performance. Como última sugestão de pesquisa futura, pode-se implementar a classificação das laranjas em um streaming para verificar a aplicação em um ambiente onde a rede receberia como entrada os frames do vídeo.

REFERÊNCIAS

- ABDULLA, W. *Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow*. 2018. Disponível em: <<https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46>>. Acesso em: 18 ago. 2022. Citado 5 vezes nas páginas 26, 29, 30, 31 e 32.
- ALVES, G. *Understanding ConvNets (CNN)*. 2018. Disponível em: <<https://medium.com/neuronio/understanding-convnets-cnn-712f2afe4dd3>>. Acesso em: 20 ago. 2022. Citado na página 21.
- BALACHANDRAN, S. *Machine Learning - Going Furthur with CNN Part 2*. 2020. Disponível em: <<https://dev.to/sandeepbalachandran/machine-learning-going-furthur-with-cnn-part-2-41km>>. Acesso em: 20 ago. 2022. Citado na página 20.
- GIRSHICK, R. *Fast R-CNN*. arXiv, 2015. Disponível em: <<https://arxiv.org/abs/1504.08083>>. Citado na página 32.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado na página 16.
- HE, K. et al. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, p. 2980–2988, 2017. Disponível em: <<https://arxiv.org/pdf/1703.06870.pdf>>. Citado na página 28.
- HIJAZI, S.; KUMAR, R.; ROWEN, C. *Using Convolutional Neural Networks for Image Recognition*. 2015. Disponível em: <http://www.multimediacdocs.com/assets/cadence_e mea/documents/using_convolutional_neural_networks_for_image_recognition.pdf>. Acesso em: 21 ago. 2022. Citado 2 vezes nas páginas 23 e 24.
- HONDA, H. *Digging into Detectron 2*. 2020. Disponível em: <<https://medium.com/@hirotooschwert/digging-into-detectron-2-47b2e794fabd>>. Acesso em: 28 ago. 2022. Citado na página 29.
- KAEHLER, A.; BRADSKI, G. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, 2017. ISBN 9781491937990. Disponível em: <<https://books.google.com.br/books?id=c7UXswEACAAJ>>. Citado 2 vezes nas páginas 16 e 17.
- LISBINSKI, F. C. et al. Perspectivas e Desafios da Agricultura 4.0 para o Setor Agrícola. *VIII Simpósio da Ciência do Agronegócio*, 2020. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/218601/001122708.pdf?sequence=1>>. Citado na página 13.
- MASSRUHÁ, S. M. F. S.; LEITE, M. A. de A. *Agro 4.0 - Rumo à Agricultura Digital*. Embrapa, 2017. Disponível em: <<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/166203/1/PL-Agro4.0-JC-na-Escola.pdf>>. Citado na página 13.
- NATIONS, D. of E. S. A. U. *World Population Prospects 2022: Data Sources*. 2021. Disponível em: <<https://population.un.org/dataportal/data/indicators/49/locations/900/start/1950/end/2100/line/linetimeplotsingle>>. Acesso em: 02 ago. 2022. Citado na página 13.

NETTO, F.; BONHO, S. *ProjectOrange*. 2021. Disponível em: <https://github.com/MiniHulk1183/project_orange>. Acesso em: 29 ago. 2022. Citado na página 36.

NIELSEN, M. A. *Neural Networks and Deep Learning*. Determination Press, 2018. Disponível em: <<http://neuralnetworksanddeeplearning.com/>>. Acesso em: 17 ago. 2022. Citado na página 18.

PATTERSON, J.; GIBSON, A. *Deep Learning: A Practitioner's Approach*. O'Reilly, 2017. ISBN 978-1-4919-1425-0. Disponível em: <<https://www.safaribooksonline.com/library/view/deep-learning/9781491924570/>>. Citado 8 vezes nas páginas 17, 18, 19, 20, 21, 22, 23 e 25.

PAYNE, A. et al. Estimation of mango crop yield using image analysis – segmentation method. *Computers and Electronics in Agriculture*, v. 91, p. 57–64, 2013. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169912002669>>. Citado na página 14.

RONG, D.; RAO, X.; YING, Y. Computer vision detection of surface defect on oranges by means of a sliding comparison window local segmentation algorithm. *Computers and Electronics in Agriculture*, v. 137, p. 59–68, 2017. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169916305579>>. Citado na página 13.

SAMUEL, A. L. *Some Studies in Machine Learning Using the Game of Checkers*. 1959. Disponível em: <<https://www.cs.virginia.edu/~evans/greatworks/samuel1959.pdf>>. Acesso em: 15 ago. 2022. Citado na página 16.

SILVA, M. V. L. da; BONHO, S. *Implementação com Detectron2*. 2022. Disponível em: <https://github.com/marcosleal/tcc_engEletronica/blob/main/Detectron2_laranjas.ipynb>. Acesso em: 9 dez. 2022. Citado na página 44.

SILVA, M. V. L. da; BONHO, S. *Mask R-CNN Image Segmentation*. 2022. Disponível em: <https://github.com/marcosleal/tcc_engEletronica/blob/main/Mask_R_CNN_Image_Segmentation.ipynb>. Acesso em: 10 dez. 2022. Citado na página 39.

WANG, C. et al. A robust fruit image segmentation algorithm against varying illumination for vision system of fruit harvesting robot. *Optik*, v. 131, p. 626–631, 2017. ISSN 0030-4026. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0030402616315133>>. Citado na página 14.

WANGENHEIM, A. von. *Deep Learning::Segmentação con CNNs*. 2018. Disponível em: <<https://lapix.ufsc.br/ensino/visao/visao-computacionaldeep-learning/deep-learningsegmentacao-semantic/>>. Acesso em: 13 ago. 2022. Citado na página 27.

WU, Y. et al. *Detectron2: A PyTorch-based modular object detection library*. 2019. Disponível em: <<https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library/>>. Acesso em: 23 ago. 2022. Citado 2 vezes nas páginas 33 e 34.

ZAPAROLLI, D. *Agricultura 4.0*. 2020. Disponível em: <<https://revistapesquisa.fapesp.br/agricultura-4-0/>>. Acesso em: 10 ago. 2022. Citado na página 13.