



UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

DANILO SOUZA DE ASSUNÇÃO

**Classificação automática de espécies de abelhas utilizando rede neural
convolucional com técnicas de aumento de dados e transferência de
aprendizado**

São Paulo

2022

DANILO SOUZA DE ASSUNÇÃO

**Classificação automática de espécies de abelhas utilizando rede neural
convolucional com técnicas de aumento de dados e transferência de
aprendizado**

Versão corrigida

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Metodologia e Técnicas da Computação

Orientador: Prof. Dr. Helton Hideraldo Bis-
caro

Coorientador: Prof. Dr. Luciano Antonio Di-
giampietri

São Paulo

2022

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Ficha catalográfica elaborada pela Biblioteca da Escola de Artes, Ciências e Humanidades,
com os dados inseridos pelo(a) autor(a)
Brenda Fontes Malheiros de Castro CRB 8-7012; Sandra Tokarevicz CRB 8-4936

Souza de Assunção, Danilo
Classificação automática de espécies de abelhas
utilizando rede neural convolucional com técnicas
de aumento de dados e transferência de aprendizado
/ Danilo Souza de Assunção; orientador, Helton
Hideraldo Biscaro; coorientador, Luciano Antonio
Digiampietri. -- São Paulo, 2022.
118 p: il.

Dissertacao (Mestrado em Ciencias) - Programa de
Pós-Graduação em Sistemas de Informação, Escola de
Artes, Ciências e Humanidades, Universidade de São
Paulo, 2022.

Versão corrigida

1. Reconhecimento de espécies de abelhas. 2.
CNN. 3. Redes neurais convolucionais. 4. Aumento de
dados. 5. Transferência de aprendizado. I. Biscaro,
Helton Hideraldo, orient. II. Digiampietri, Luciano
Antonio, coorient. III. Título.

Dissertação de autoria de Danilo Souza de Assunção, sob o título “**Classificação automática de espécies de abelhas utilizando rede neural convolucional com técnicas de aumento de dados e transferência de aprendizado**”, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em 06 de outubro de 2022 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Helton Hideraldo Biscaro
Universidade de São Paulo
Presidente

Prof. Dr. Roberto Hirata Junior
Universidade de São Paulo

Prof. Dr. Tiago Mauricio Francoy
Universidade de São Paulo

Dedico este trabalho aos meus familiares, em especial a minha mãe Alessandra Marcia de Souza que esteve sempre ao meu lado me apoiando nesta trajetória.

Agradecimentos

Primeiramente, eu desejo agradecer à minha mãe, Alessandra Marcia de Souza, sendo a minha melhor companhia em todo o trajeto pelo qual eu percorri neste mestrado, ela esteve comigo, me incentivando e apoiando, e isso tudo foi concluído graças à força que ela constantemente me forneceu.

À minha família, pai, irmãos, tios e avós que também me apoiaram bastante durante todo este percurso, principalmente ao meu tio, William Assunção, sendo uma das pessoas que mais me induziram a entrar no meio acadêmico.

Um reconhecimento especial aos meus orientadores, Prof. Dr. Helton Hideraldo Biscaro e Prof. Dr. Luciano Antonio Digiampietri que foram figuras excepcionais para mim, ao Helton por ter me aceitado como um dos seus orientandos, me ajudando muito durante todo o desenvolvimento do trabalho e aos ensinamentos por ele dados a mim, e ao Luciano que foi o facilitador em todo o meu processo de entrada neste programa de mestrado e também uma das pessoas que mais me auxiliou durante todo este percurso, e a estes grandes mestres, meu eterno agradecimento.

Ao meu amigo Gilson Clemente, que me ajudou muito durante todo o desenvolvimento do meu trabalho, tendo participado em opiniões gerais no escopo presente neste documento, como também um grande revisor. E também um agradecimento especial à Sammy, André e Anne, que são pessoas muito importantes para mim e que também me apoiaram muito.

E por fim, ao PPGSI, sendo o programa que tornou tudo isto possível, permitindo que eu participasse deste grande percurso, aprendi muito, e aqui estou, muito mais avançado do que quando comecei. Para todos os colaboradores que fazem parte deste programa, meus agradecimentos.

“Nada é permanente, exceto a mudança.”

(Heráclito)

Resumo

ASSUNÇÃO, Danilo Souza. **Classificação automática de espécies de abelhas utilizando rede neural convolucional com técnicas de aumento de dados e transferência de aprendizado**. 2022. 118 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2022.

O processo de classificação de espécies de abelhas é uma atividade importante para a preservação das abelhas, pois possibilita o estabelecimento de estratégias mais precisas de conservação ao obter informações detalhadas de uma determinada espécie numa localidade específica. A realização automática da análise estatística da morfologia das abelhas se convém devido às despesas necessárias nos processos de identificação manual, e, para esta análise morfológica, o conjunto de características extraídos a partir das asas têm se mostrado uma eficiente maneira para a identificação das espécies utilizando métodos estatísticos ou computacionais. O aprendizado profundo tem sido amplamente aplicado em atividades relacionadas à visão computacional. Diferente dos métodos tradicionais de aprendizado de máquina, o modelo de aprendizado profundo pode aprender características automaticamente a partir de uma grande quantidade de instâncias de dados, e não requer a assistência de um especialista no domínio para a extração destas características. No aprendizado profundo, as Redes Neurais Convolucionais (CNN) são bem conhecidas por seu sucesso em muitas tarefas de visão computacional. Neste trabalho foi desenvolvido um método utilizando CNN para classificação de espécies de abelhas através da morfologia de suas asas através do auxílio de técnicas de aumento de dados e transferência de aprendizado. A solução desenvolvida atingiu resultados superiores a 97% para acurácia e medida F.

Palavras-chaves: Redes Neurais Convolucionais. CNN. Aumento de dados. Transferência de aprendizado. Reconhecimento de espécies. Reconhecimento de espécies de abelhas. Morfologia das asas. Reconhecimento por imagens digitais.

Abstract

ASSUNÇÃO, Danilo Souza. **Automatic Classification of Bee Species using CNN with Data Augmentation and Transfer Learning Techniques**. 2022. 118 p.
Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2022.

Bees' species classification process turns out to be an important activity in bee preservation, as it makes it possible to establish more precise conservation strategies when obtaining detailed information about a specific species and also for a specific location. The automatic realization of the visual analysis of the morphology of the bees is convenient due to the necessary expenses in the processes of manual identification, and, for this morphological analysis, the sets of characteristics extracted from the wings have shown to be an efficient way for the identification of the species using statistical methods. Deep learning has been widely applied in activities related to computer vision. Unlike traditional machine learning methods, the deep learning model can automatically learn resources from a large number of data samples and does not require the assistance of an expert in the field to extract these characteristics. In deep learning, Convolutional Neural Networks (CNN) are well known for their success in many computer vision tasks. In this work, a method was developed using CNN to classify bee species through the morphology of their wings using data augmentation and transfer learning techniques. The developed solution achieved results above 97% for accuracy and F score.

Keywords: Convolutional Neural Networks. CNN. Data augmentation. Transfer learning. Species recognition. Recognition of bee species. Wing morphology. Digital image recognition.

Lista de figuras

Figura 1 – Representação gráfica de subajuste e sobreajuste	28
Figura 2 – Arquitetura simplificada de uma rede neural artificial	31
Figura 3 – Arquitetura macro de uma CNN	32
Figura 4 – Processo da convolução em dados de entrada	33
Figura 5 – Exemplo do uso do passo de valor 2 no processo convolucional	34
Figura 6 – Exemplo do uso de preenchimento de valor 0 no processo convolucional	35
Figura 7 – Agrupamento máximo	36
Figura 8 – Agrupamento médio	37
Figura 9 – Diagrama do processo de revisão sistemática	43
Figura 10 – Ano de publicação dos artigos com base em palavras chaves	50
Figura 11 – Ano de publicação dos artigos selecionados	51
Figura 12 – Técnicas de pré-processamento de imagens aplicadas	52
Figura 13 – Técnicas de Transferência de aprendizado	53
Figura 14 – Métodos para o aumento de dados	54
Figura 15 – Arquitetura de CNNs	56
Figura 16 – Tipos agrupados de arquiteturas de CNNs	57
Figura 17 – Método de classificação utilizado	58
Figura 18 – Asa de abelha da espécie <i>Trigona spinipes</i>	60
Figura 19 – Representação do módulo do conjunto de dados	63
Figura 20 – Processamento e organização das imagens	64
Figura 21 – Divisão dos conjuntos de dados	64
Figura 22 – Aumento do conjunto de dados de treinamento	65
Figura 23 – Compactação e armazenamento do conjunto de dados	66
Figura 24 – Representação do módulo do modelo de classificação	66
Figura 25 – Grupo de configuração	67
Figura 26 – Grupo de definição do modelo	68
Figura 27 – Grupo de treinamento e validação	69
Figura 28 – Grupo de teste	69
Figura 29 – Estratégia de experimentação	70
Figura 30 – Resultados do treinamento com a InceptionResNet v2 + Dados originais	73

Figura 31 – Resultados do treinamento da InceptionResNet v2 + Dados aumentados	75
Figura 32 – Resultados do treinamento com a InceptionResNet v2 + Dados originais + Transferência de aprendizado	76
Figura 33 – Resultados do treinamento com a InceptionResNet v2 + Dados aumentados + Transferência de aprendizado	77
Figura 34 – Resultados do treinamento com a Inception v3 + Dados originais . . .	79
Figura 35 – Resultados do treinamento da Inception v3 + Dados aumentados . . .	80
Figura 36 – Resultados do treinamento com a Inception v3 + Dados originais + Transferência de aprendizado	81
Figura 37 – Resultados do treinamento com a Inception v3 + Dados aumentados + Transferência de aprendizado	82
Figura 38 – Resultados do treinamento com a ResNet-152 + Dados originais	84
Figura 39 – Resultados do treinamento da ResNet-152 + Dados aumentados	85
Figura 40 – Resultados do treinamento com a ResNet-152 + Dados originais + Transferência de aprendizado	86
Figura 41 – Resultados do treinamento com a ResNet-152 + Dados aumentados + Transferência de aprendizado	87
Figura 42 – Resultados do treinamento com a VGG16 + Dados originais	88
Figura 43 – Resultados do treinamento da VGG16 + Dados aumentados	89
Figura 44 – Resultados do treinamento com a VGG16 + Dados originais + Transferência de aprendizado	90
Figura 45 – Resultados do treinamento com a VGG16 + Dados aumentados + Transferência de aprendizado	91
Figura 46 – Resultados do treinamento com a VGG19 + Dados originais	93
Figura 47 – Resultados do treinamento da VGG19 + Dados aumentados	94
Figura 48 – Resultados do treinamento com a VGG19 + Dados originais + Transferência de aprendizado	95
Figura 49 – Resultados do treinamento com a VGG19 + Dados aumentados + Transferência de aprendizado	96
Figura 50 – Resultados do treinamento usando Adagrad	99
Figura 51 – Resultados do treinamento usando Adadelta	100
Figura 52 – Resultados do treinamento usando Adamax	102
Figura 53 – Média aritmética de acurácia e medida F por Grupo	104

Figura 54 – Média de Épocas por Grupo 105

Lista de quadros

Quadro 1 – Resultados do teste com a InceptionResNet v2 + Dados originais . . .	74
Quadro 2 – Resultados do teste com a InceptionResNet v2 + Dados aumentados .	75
Quadro 3 – Resultados do teste com a InceptionResNet v2 + Dados originais + Transferência de aprendizado	76
Quadro 4 – Resultados do teste com a InceptionResNet v2 + Dados aumentados + Transferência de aprendizado	77
Quadro 5 – Resultados do teste com a Inception v3 + Dados originais	79
Quadro 6 – Resultados do teste com a Inception v3 + Dados aumentados	80
Quadro 7 – Resultados do teste com a Inception v3 + Dados originais + Trans- ferência de aprendizado	81
Quadro 8 – Resultados do teste com a Inception v3 + Dados aumentados + Trans- ferência de aprendizado	82
Quadro 9 – Resultados do teste com a ResNet-152 + Dados originais	84
Quadro 10 – Resultados do teste com a ResNet-152 + Dados aumentados	85
Quadro 11 – Resultados do teste com a ResNet-152 + Dados originais + Trans- ferência de aprendizado	86
Quadro 12 – Resultados do teste com a ResNet-152 + Dados aumentados + Trans- ferência de aprendizado	87
Quadro 13 – Resultados do teste com a VGG16 + Dados originais	89
Quadro 14 – Resultados do teste com a VGG16 + Dados aumentados	89
Quadro 15 – Resultados do teste com a VGG16 + Dados originais + Transferência de aprendizado	91
Quadro 16 – Resultados do teste com a VGG16 + Dados aumentados + Trans- ferência de aprendizado	91
Quadro 17 – Resultados do teste com a VGG19 + Dados originais	93
Quadro 18 – Resultados do teste com a VGG19 + Dados aumentados	94
Quadro 19 – Resultados do teste com a VGG19 + Dados originais + Transferência de aprendizado	95
Quadro 20 – Resultados do teste com a VGG19 + Dados aumentados + Trans- ferência de aprendizado	96

Quadro 21 – Resultados do teste usando	99
Quadro 22 – Resultados do teste usando Adadelta	101
Quadro 23 – Resultados do teste usando Adamax	102

Lista de tabelas

Tabela 1 – Sumário de todos os artigos analisados	46
Tabela 2 – Conjunto de dados original das asas de abelhas	61

Lista de abreviaturas e siglas

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
GAN	Generative Adversarial Networks
KNN	K-Nearest Neighbors
MNIST	Modified National Institute of Standards and Technology
RBF	Radius Basis Function
ReLU	Rectified Linear Unit
RGB	Red, Green and Blue
ResNet	Residual Neural Network
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
VGG	Visual Geometry Group

Sumário

1	Introdução	19
1.1	<i>Hipótese</i>	21
1.2	<i>Objetivos</i>	21
1.3	<i>Justificativa</i>	22
1.4	<i>Estrutura do documento</i>	22
2	Conceitos fundamentais	23
2.1	<i>Visão computacional</i>	23
2.2	<i>Aprendizado de máquina</i>	24
2.2.1	Principais tarefas em aplicações de aprendizado de máquina	24
2.2.2	Terminologias	25
2.2.3	Aprendizagem supervisionada	26
2.2.4	Sobreajuste e subajuste	27
2.2.5	Avaliação de desempenho de classificadores	28
2.3	<i>Aprendizado profundo</i>	29
2.3.1	Algoritmo de otimização	30
2.4	<i>Redes neural artificial</i>	30
2.5	<i>Redes Neurais Convolucionais</i>	31
2.5.1	Camada convolucional	32
2.5.2	Função de ativação não linear	35
2.5.3	Camada de agrupamento	36
2.5.4	Camada totalmente conectada	37
2.6	<i>Técnica de transferência de aprendizado</i>	37
2.7	<i>Técnica de aumento de dados</i>	38
2.8	<i>Conclusão</i>	40
3	Revisão sistemática	41
3.1	<i>Objetivos da revisão</i>	41
3.2	<i>Protocolo</i>	42
3.3	<i>Resultado da revisão sistemática</i>	45
3.3.1	Ano de publicação	50

3.3.2	Técnicas de pré-processamento de imagens	52
3.3.3	Técnica de transferência de aprendizado	53
3.3.4	Técnicas de aumento de dados	54
3.3.5	Modelo de arquitetura da CNN	55
3.3.6	Métodos de classificação	58
3.4	<i>Conclusão</i>	59
4	 Materiais e métodos	60
4.1	<i>Módulo do conjunto de dados</i>	63
4.2	<i>Módulo do modelo de classificação</i>	66
4.2.1	Grupo de configuração	67
4.2.2	Grupo de definição do modelo	68
4.2.3	Grupo de treinamento e validação	68
4.2.4	Grupo de teste	69
4.3	<i>Avaliação</i>	70
5	 Resultados e discussões	72
5.1	<i>Seleção da arquitetura da CNN</i>	72
5.1.1	InceptionResNet v2	72
5.1.2	Inception v3	78
5.1.3	ResNet-152	83
5.1.4	VGG16	88
5.1.5	VGG19	92
5.1.6	Conclusão	97
5.2	<i>Seleção do algoritmo de otimização</i>	98
5.2.1	Adagrad	98
5.2.2	Adadelta	100
5.2.3	Adamax	101
5.2.4	Conclusão	102
5.3	<i>Modelo final</i>	103
5.4	<i>Conclusão dos resultados</i>	104
6	 Conclusão	107
6.1	<i>Limitações</i>	108

6.2	<i>Trabalhos futuros</i>	108
	REFERÊNCIAS	110
	Apêndice A – Matriz confusão do modelo final	116
	Apêndice B – Medidas obtidas no modelo final por espécie .	117
	Apêndice C – Medidas obtidas no modelo final por gênero de espécies	118

1 Introdução

A partir da definição dos serviços ecossistêmicos, que designa sobre o funcionamento normal dos ecossistemas e os benefícios que as pessoas adquirem dele, tem-se uma conexão direta com a prosperidade da civilização humana. Tendo como essencialidade a polinização, sendo um dos serviços ecossistêmicos que traz segurança alimentar, mas é afetada desde a Revolução Industrial com as enormes perdas de biodiversidades que o mundo vem vivenciando (ROCKSTRÖM *et al.*, 2009). O valor de mercado anual dos serviços de polinização foi estimado em US\$ 235 bilhões em todo o mundo (POTTS *et al.*, 2016). Sendo as abelhas as polinizadoras primárias de culturas e ecossistemas naturais e, responsáveis pela reprodução da grande maioria das plantas com flores, tornando-as organismos essenciais tanto em termos econômicos quanto ecológicos, além de serem vitais para fins de conservação (REBELO *et al.*, 2021).

Diferentes abordagens automáticas têm sido desenvolvidas com o objetivo de identificar automaticamente a espécie de diferentes seres vivos de forma a auxiliar a minimizar os problemas relacionados com os gastos econômicos e, principalmente, pelas perdas de biodiversidades dos ecossistemas. Estas abordagens envolvem, por exemplo, o código de barras de DNA e o reconhecimento automático de espécies com base na morfologia das asas, sendo que, estas metodologias são vantajosas por permitirem o acesso à identificação de espécies a um público mais amplo, deixando os taxonomistas despendidos em tarefas mais urgentes, como a descrição de espécies (REBELO *et al.*, 2021). Vale ressaltar que, para algumas espécies, a identificação automática pode ser um desafio, principalmente para as abelhas sem ferrão que possuem características muito parecidas entre as espécies de mesmo grupo (FRANCOY; IMPERATRIZ-FONSECA, 2010)

Atualmente, a realização automática da análise visual da morfologia das abelhas é mais proveitosa diante das dificuldades encontradas nos processos de identificação manual, e para tal automatização da análise morfológica, os conjuntos de características extraídos a partir das asas têm se mostrado uma maneira eficiente para a identificação das espécies utilizando métodos estatísticos (FRANCOY *et al.*, 2008).

Baseado no problema abordado, em conjunto com suas possíveis soluções, alguns experimentos utilizando métodos tradicionais de aprendizado de máquina já foram realizados no trabalho de Rebelo *et al.* (2021), onde é feito o uso de um conjunto de técnicas de

pré-processamento e segmentação de imagens, podendo ser denominados como métodos *handcrafted*, para a extração manual de características relevantes e situadas nas imagens de asas de abelhas, para que, assim, fosse criado um conjunto de imagens com características capazes de serem usadas em um modelo de aprendizado de máquina supervisionado utilizando o algoritmo *k-nearest neighbors* (KNN).

A partir dos resultados do experimento citado, o presente projeto objetivou realizar um aprimoramento do mecanismo que realiza todo o processo de identificação automática das espécies de abelhas através das imagens de suas asas. Desta forma, foi idealizado o uso métodos de aprendizado profundo, ou, mais especificamente, o uso de Redes Neurais Convolucionais (CNN) para a construção de ferramentas que permitam classificar automaticamente tais espécies, removendo a necessidade do uso de métodos *handcrafted* para a extração de características contidas nessas imagens, automatizando por completo o fluxo.

Neste contexto, na literatura, foi identificado que o aprendizado profundo tem sido amplamente utilizado em atividades relacionadas à visão computacional. O aprendizado profundo é um subconjunto do aprendizado de máquina baseado em redes neurais artificiais. Este processo de aprendizado é profundo pois a estrutura das redes neurais artificiais consiste em várias camadas, sendo estas camadas, de entrada, saída e oculta. Diferente dos métodos tradicionais de aprendizado de máquina, o modelo de aprendizado profundo pode aprender características automaticamente a partir de uma grande quantidade de dados, e não requer a assistência de um especialista no domínio para extração manual destas características, dado que, isso é extraído automaticamente através do algoritmo (LIU *et al.*, 2020b). No campo de aprendizado profundo, as CNNs são bem reconhecidas por seu sucesso em muitas tarefas de visão computacional (LE *et al.*, 2020).

A CNN oferece uma abordagem que automatiza o aprendizado de características contidas em uma imagem ao utilizar grandes conjuntos de dados, que representam um domínio de interesse da aplicação (GONZALEZ, 2018). Devido à necessidade de uma grande quantidade de dados para o treinamento deste tipo de rede, é essencial que, para conjuntos de dados menores, tenha-se o auxílio de técnicas que permitam atingir bons resultados. Baseado nisso, foram adicionados, no presente trabalho, mecanismos de aumento de dados e/ou transferência de aprendizado, com o propósito de se obter um modelo com melhor desempenho, por meio do uso de uma CNN que irá identificar, a partir da disponibilidade de um pequeno conjunto de dados das asas de abelhas, de maneira automatizada e sem o uso de métodos *handcrafted*.

1.1 Hipótese

O uso de CNN para automatização do processo de extração de características e classificação, pode trazer resultados promissores ao problema em questão, devido a sua capacidade de identificar e extrair características de alto e baixo nível, presente nas imagens. E, estes resultados podem ser auxiliados baseando-se no uso da CNN em união com as técnicas de aumento de dados e/ou técnicas de transferência de aprendizado.

Esta hipótese foi testada, levando em consideração, as medidas tradicionais de aprendizado de máquina obtidas a partir do modelo proposto, relacionando-o com o modelo de *baseline* apresentado em [Rebello et al. \(2021\)](#).

Espera-se que o modelo proposto apresente resultados similares ou superiores ao modelo de *baseline* para as medidas sugeridas.

1.2 Objetivos

Este projeto de pesquisa tem como objetivo o desenvolvimento de uma ferramenta de classificação de espécies de abelhas por meio da morfologia de suas asas, utilizando-se de uma CNN como mecanismo de extração de características e classificação destas imagens, junto à utilização de técnicas de aumento de dados ou transferência de aprendizado para otimização dos resultados do modelo, a fim de se obter resultados relevantes, como apresentados no *baseline* definido. Desta forma, este trabalho tem os seguintes objetivos específicos:

1. Desenvolvimento, teste e validação de ferramenta (*pipeline*) que realize automaticamente o processamento, ampliação, organização e armazenamento de conjunto de dados contendo as imagens para treinamento, validação e teste do modelo;
2. Desenvolvimento, teste e validação de ferramenta (*pipeline*) que utilize o conjunto de dados processado e disponibilizado em união a arquiteturas de CNN para experimentação dos modelos no domínio proposto.

1.3 Justificativa

Uma vez que o modelo demonstrar cenários de sucesso com base nas avaliações e resultados extraídos, torna-se evidente que a aplicação de uma CNN poderá atingir um modelo mais próximo do estado da arte para o problema abordado.

1.4 Estrutura do documento

O restante deste documento está organizado da seguinte forma. O capítulo 2 descreve os conceitos fundamentais para o entendimento do trabalho. O capítulo 3 apresenta a revisão de literatura realizada. O capítulo 4 descreve o conjunto de dados e as técnicas a serem utilizadas. O capítulo 5 detalha a abordagem desenvolvida. O capítulo 6 apresenta os resultados e as discussões realizadas em cima destes resultados. E por fim, o capítulo 7 apresenta a conclusão deste projeto e suas possibilidades futuras.

2 Conceitos fundamentais

Neste capítulo são apresentados alguns tópicos fundamentais para a compreensão deste projeto. Dessa forma, primeiro é realizada uma curta introdução ao tema de visão computacional, logo depois são apresentadas as definições referentes a aprendizado de máquina, aprendizado profundo, redes neurais artificiais, Redes Neurais Convolucionais (CNN), técnicas de transferência de aprendizado e aumento de dados.

2.1 Visão computacional

A Visão Computacional pode ser descrita como uma subárea da Computação Gráfica, a qual busca extrair informações relevantes das imagens. Em outras palavras, a entrada de dados é uma imagem e a saída são informações como: existe ou não determinado objeto, características ou atividade na imagem de entrada (REBELO *et al.*, 2021). E a meta da visão computacional é utilizar computadores para simular a visão humana, incluindo o aprendizado e a capacidade de fazer inferências e agir com base em informações visuais. Essa área representa um ramo da inteligência artificial cujo objetivo é simular a inteligência humana (GONZALEZ; WOODS, 2000).

Neste contexto, conforme a definição dada por Gonzalez e Woods (2000), uma imagem pode ser estabelecida como uma função bidimensional, $f(x, y)$, em que x e y são coordenadas espaciais (plano), e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou nível de cinza da imagem nesse ponto. Quando x , y e os valores de intensidade de f são quantidades finitas e discretas, chamamos de imagem digital. O campo do processamento digital de imagens se refere ao processamento de imagens digitais por um computador digital. Observe que uma imagem digital é composta de um número finito de elementos, cada um com localização e valor específicos. Esses elementos são chamados de elementos *pictóricos*, elementos de imagem, *pels* e *pixels*. *Pixel* é o termo mais utilizado para representar os elementos de uma imagem digital.

Para um modelo de visão computacional, tem-se, conforme demonstrado por Gonzalez (2018), a composição de determinadas etapas, sendo tais etapas, aquisição, pré-processamento, extração de características e classificação. A aquisição gera os padrões brutos de entrada (por exemplo, imagens digitais); o pré-processamento trata de tarefas

como redução de ruído e correções geométricas; a extração de características lida com atributos computacionais que são fundamentais para diferenciar uma classe de padrões de outra; e a classificação é o processo que atribui um determinado padrão de entrada a uma das várias classes predefinidas.

2.2 *Aprendizado de máquina*

Dentre as definições presentes na literatura, [Mohri, Rostamizadeh e Talwalkar \(2018\)](#) dizem que, o aprendizado de máquina pode ser amplamente definido como métodos computacionais, usando a experiência (dados) para melhorar o desempenho ou realizar previsões precisas. A experiência se refere às informações anteriores disponíveis para o modelo de aprendizagem de máquina, que normalmente assumem a forma de dados eletrônicos coletados e disponibilizados para análise. Esses dados podem estar na forma de conjuntos de treinamento com identificação humana ou a diferentes tipos de informações obtidas por meio da interação com o ambiente. Em todos os casos, sua qualidade e tamanho são cruciais para o sucesso das previsões feitas pelo modelo. De maneira geral, as técnicas de aprendizagem são métodos orientados por dados que combinam conceitos fundamentais em ciência da computação com ideias de estatística, probabilidade e otimização.

2.2.1 Principais tarefas em aplicações de aprendizado de máquina

Nesta seção são explicados alguns conceitos sobre as tarefas de aprendizado de máquina, sendo comumente utilizados e explicados em [Mohri, Rostamizadeh e Talwalkar \(2018\)](#). Tais tarefas de aprendizado de máquina são:

- **Classificação:** trata-se do problema de atribuir uma categoria a cada item. Por exemplo, a classificação de imagens consiste em atribuir a cada imagem uma categoria como carro, navio ou avião. O número de categorias em tais tarefas costuma ser menor do que algumas centenas, mas pode ser muito maior em algumas tarefas difíceis, e até mesmo ilimitadas, como em classificação de texto ou reconhecimento de fala.
- **Regressão:** é a atividade de solucionar problemas já previstos a partir de dados históricos de cada item. Exemplos de regressão incluem a previsão de valores de

estoque ou índice econômicos. Na regressão, a penalidade para uma previsão incorreta depende da magnitude da diferença entre os valores verdadeiros e previstos, em contraste com o problema de classificação, em que normalmente não há noção de proximidade entre as várias categorias.

- **Redução de dimensionalidade:** consiste em transformar uma representação inicial de itens em uma representação de dimensão inferior, preservando algumas propriedades da representação inicial. Um exemplo comum envolve o pré-processamento de imagens digitais em tarefas de visão computacional.

2.2.2 Terminologias

Baseado nas terminologias empregadas em [Mohri, Rostamizadeh e Talwalkar \(2018\)](#), a seguir está uma lista de definições e terminologias habitualmente usados no escopo de aprendizado de máquina:

- **Instâncias:** são unidades que representam cada dado dentro de um conjunto de dados que são usados para aprendizado e avaliação do modelo. Imaginando o problema de identificação das espécies de abelhas através da imagem de suas asas, uma instância seria o correspondente a uma imagem de uma asa de abelha dentro do conjunto de imagens presentes no conjunto de dados.
- **Características:** O conjunto de atributos, frequentemente representado como um vetor, associado a uma instância. No caso de uma imagem, algumas características relevantes podem incluir a forma do objeto presente nesta imagem, cor e dentre vários outros aspectos.
- **Rótulo ou Classe:** valores ou categorias atribuídos a exemplos. Em problemas de classificação, as instâncias são atribuídas a categorias específicas. Por exemplo, considerando imagens de objetos de veículos, se tem categorias como, carro, avião ou navio.
- **Hiperparâmetros:** parâmetros livres que não são determinados pelo algoritmo de aprendizagem, mas sim especificados como entrada para o algoritmo de aprendizagem.
- **Instância de treinamento:** são instâncias utilizadas para treinar um algoritmo de aprendizagem. Em um problema envolvendo reconhecimento de padrões por meio de

imagens digitais, a instância de treinamento consiste em um conjunto de exemplos de imagens junto a suas classes associadas.

- **Instância de validação:** são exemplos usados para ajustar os parâmetros de um algoritmo de aprendizagem ao trabalhar com dados rotulados. A instância de validação é usada para selecionar valores apropriados para os parâmetros livres do algoritmo de aprendizagem (hiperparâmetros).
- **Instância de teste:** são exemplos usados para avaliar o desempenho de um algoritmo de aprendizagem. A instância de teste é separada dos dados de treinamento e validação, não sendo disponibilizada na fase de aprendizado. Por exemplo, em um cenário de reconhecimento facial, a instância de teste consiste em uma coleção de exemplos de imagens de rostos, para os quais o algoritmo de aprendizagem deve prever suas determinadas classes com base em certas características, sendo então comparadas com as características da instância de teste para medir o desempenho do algoritmo.
- **Função de perda:** em problemas de otimização, a função de perda é um tipo de função que mede o grau de previsões incorretas realizadas pelo algoritmo, ou seja, se o algoritmo apresentar uma qualidade não desejada, o valor exibido pela função de perda será alto.

2.2.3 Aprendizagem supervisionada

A aprendizagem supervisionada é um ramo do aprendizado de máquina, método de análise de dados que usa algoritmos que aprendem iterativamente a partir dos dados para permitir que os computadores encontrem informações ocultas sem serem explicitamente programados para onde procurar (O'SHEA; NASH, 2015). A aprendizagem supervisionada resolve problemas já conhecidos usando um conjunto de dados previamente rotulado para treinar um algoritmo para realizar tarefas específicas (NASTESKI, 2017). Ela usa modelos para prever resultados conhecidos, por exemplo, um processo de aprendizagem supervisionada poderia ser o de classificar veículos de duas e quatro rodas a partir de suas imagens. Os dados de treinamento teriam que ser rotulados corretamente para identificar se um veículo é de duas rodas ou de quatro rodas. A aprendizagem supervisionada permite

que os algoritmos “aprendam” a partir de dados históricos de treinamento e os apliquem a entradas desconhecidas para derivar a saída correta (O’SHEA; NASH, 2015).

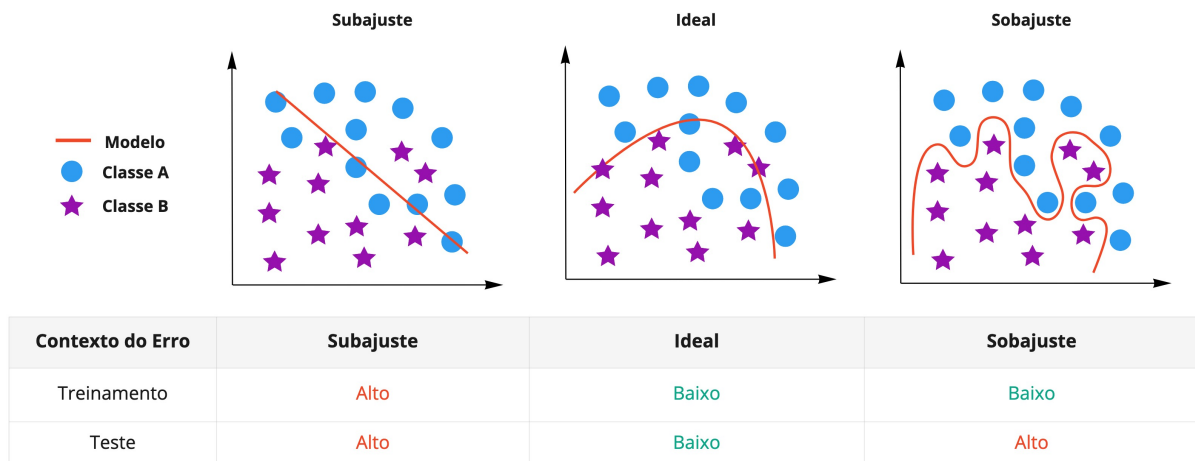
2.2.4 Sobreajuste e subajuste

Conforme apresentado em Bonaccorso (2017), o objetivo de um modelo de aprendizado de máquina é aproximar uma função desconhecida que associe elementos de entrada aos de saída (para um classificador, os chamamos de classes) . No entanto, essa condição ideal nem sempre é fácil de se alcançar, e este comportamento ocorre devido a duas características eminentes que são:

- **Sobreajuste:** ocorre quando o modelo perde sua capacidade de generalização, sendo assim, ele pode associar quase perfeitamente todas as instâncias conhecidas aos valores de saída correspondentes, mas quando uma entrada desconhecida é apresentada, o erro de previsão correspondente pode ser muito alto.
- **Subajuste:** ocorre quando o modelo não é capaz de determinar um padrão, sendo incapaz de determinar uma relação entre os dados e já possuindo resultados inconsistentes na fase de treinamento do modelo.

Para exemplificar estes conceitos, na Figura 1 temos como primeiro caso o cenário de subajuste, em que a linha vermelha retrata os resultados do modelo que indica a incapacidade de generalização com relação aos dados do exemplo, dado que, ele não é capaz de distinguir os elementos abordados. Na imagem do meio, temos o cenário ideal, apresentando a capacidade do modelo em gerar uma função capaz de se generalizar com o modelo proposto. Por fim é exibido o cenário de sobreajuste, em que é possível observar a linha vermelha definindo o resultado do modelo se ajustou exatamente aos exemplos propostos, entretanto, dado a inserção de novas instâncias, este modelo pode vir a apresentar altas taxas de erro.

Figura 1 – Representação gráfica de subajuste e sobreajuste



Fonte – Danilo Assunção, 2022

2.2.5 Avaliação de desempenho de classificadores

Ao categorizar uma instância baseada em um algoritmo de classificação, pode-se determinar que tal instância pertence a um dos quatro itens seguintes:

- Verdadeiro positivo
- Verdadeiro negativo
- Falso positivo
- Falso negativo

Um Verdadeiro positivo é um resultado em que o modelo prevê corretamente a classe positiva. Da mesma forma, um Verdadeiro negativo é um resultado em que o modelo prevê corretamente a classe negativa. Um Falso positivo é um resultado em que o modelo prevê incorretamente a classe positiva. E um Falso negativo é um resultado em que o modelo prevê incorretamente a classe negativa.

Para fins de medição de qualidade do modelo de aprendizado de máquina, são utilizadas métricas de avaliação de classificadores. As funções mais usuais entre elas são:

- Acurácia
- Precisão
- Revocação
- Medida F

Acurácia corresponde ao total de instâncias classificadas corretamente (independentemente da classe) dividido pelo número total de instâncias. Sua equação pode ser descrita pela Equação 1.

$$Acurácia = \frac{Verdadeiro\ positivo + Verdadeiro\ negativo}{Verdadeiro\ positivo + Verdadeiro\ negativo + Falso\ positivo + Falso\ negativo} \quad (1)$$

Precisão é a parcela de instâncias que obtiveram uma identificação e classificação corretamente. Podendo ser descrita a partir de seu cálculo pela Equação 2.

$$Precisão = \frac{Verdadeiro\ positivo}{Verdadeiro\ positivo + Falso\ positivo} \quad (2)$$

Revocação é a parcela de instâncias com identificação correta, em que pertencem a uma determinada classe entre todas as instâncias desta mesma classe. Seu cálculo pode ser descrito a partir da Equação 3.

$$Revocação = \frac{Verdadeiro\ positivo}{Verdadeiro\ positivo + Falso\ negativo} \quad (3)$$

Medida F corresponde à média harmônica entre a precisão e a revocação. O valor da medição torna-se ideal quanto mais próximo de 1,0 (ou 100%), enquanto que, quanto mais próximo de 0 (ou 0%), expressa um resultado ruim. Sua equação pode ser descrita pela Equação 4.

$$Medida\ F = 2 \times \frac{Precisão \times Revocação}{Precisão + Revocação} \quad (4)$$

2.3 Aprendizado profundo

Dentre as diversas definições presentes, no artigo de [Lauzon \(2012\)](#), o aprendizado profundo é sobre o aprendizado automático de vários níveis de representações da distribuição subjacente dos dados a serem modelados. Em outras palavras, um algoritmo de aprendizado profundo extrai automaticamente as características de baixo e alto nível, necessários para a classificação. Por características de alto nível, entende-se a característica que depende hierarquicamente de outras características, enquanto as características de baixo nível são detalhes únicos extraídos sucessivamente dentre as camadas da rede neural.

2.3.1 Algoritmo de otimização

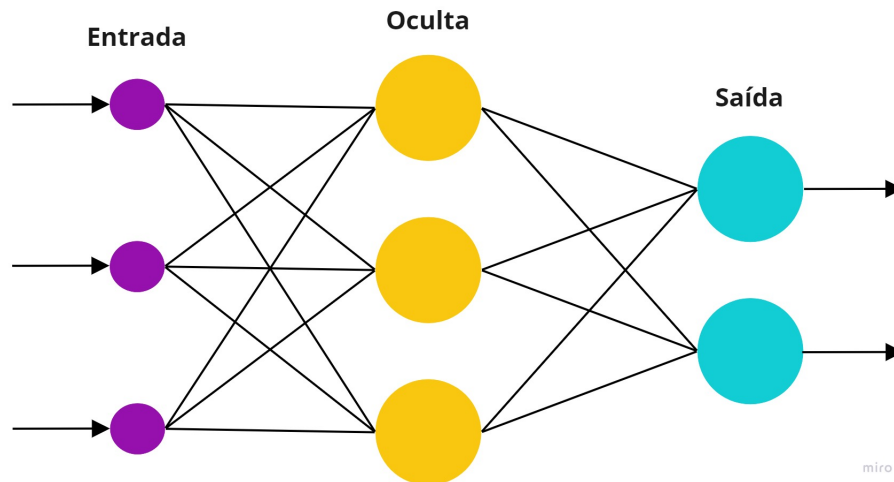
Algoritmos de otimização são métodos que possibilitam a capacidade de algoritmos de aprendizado de máquina a aprenderem através de algoritmos que computam gradientes (gradiente descendente) (ZAHEER; SHAZIYA, 2019), sendo um método que consiste em descobrir, iterativamente, os valores dos parâmetros (pesos) da rede na tentativa de minimizar a função de perda, para que assim seja maximizado o fator de previsibilidade do modelo, já que o objetivo principal deste método é diminuir o erro da rede.

No aprendizado profundo existem otimizadores de categoria adaptativa, que são denominados dessa forma pois eles usam taxas de aprendizado diferentes para cada iteração. A mudança na taxa de aprendizado depende da diferença nos parâmetros (pesos) durante o treinamento. Quanto mais os parâmetros forem alterados, menores serão as alterações na taxa de aprendizado. Essa modificação é altamente benéfica porque os conjuntos de dados do mundo real contêm características esparsas e densas, portanto, dependendo do cenário, pode não ser muito adequado ter o mesmo valor para a taxa de aprendizado para todas as características.

2.4 Redes neural artificial

As redes neurais artificiais (*artificial neural networks* - ANN) são sistemas de processamento computacional que são fortemente inspirados pelo modo como os sistemas nervosos biológicos (como o cérebro humano) operam (O'SHEA; NASH, 2015). Uma rede neural consiste em muitos elementos de processamento unidos para formar uma rede apropriada com funções de ponderação ajustáveis para cada entrada. Esses elementos de processamento geralmente são organizados em uma sequência de camadas com conexões completas ou aleatórias entre as camadas. Normalmente, existem três ou mais camadas, uma camada de entrada onde os dados são apresentados, uma ou mais camadas intermediárias ou ocultas e uma camada de saída contendo a resposta de saída para uma determinada entrada (UHRIG, 1995). A estrutura básica de uma ANN pode ser modelada como mostrado na Figura 2.

Figura 2 – Arquitetura simplificada de uma rede neural artificial

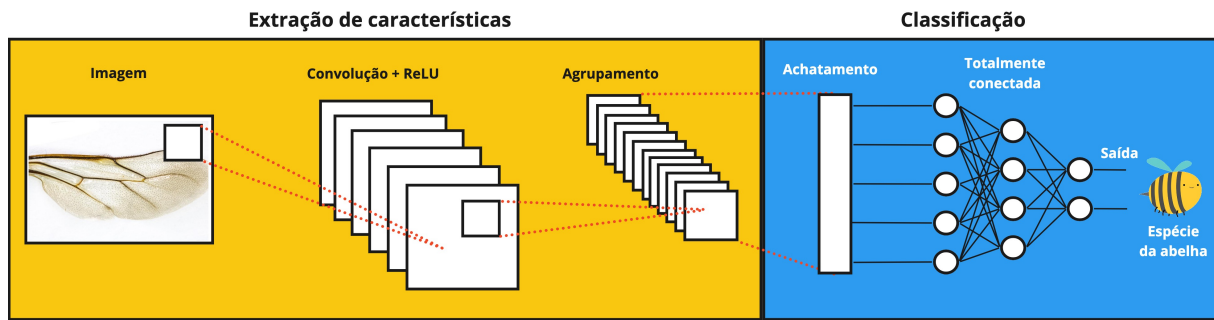


Fonte – Danilo Assunção, 2022

2.5 Redes Neurais Convolucionais

Baseado na definição em [Yamashita *et al.* \(2018\)](#), a Rede Neural Convolutiva (CNN) é uma técnica de aprendizado profundo para o processamento de dados, inspirada na organização do córtex visual animal e projetada para aprender, de forma automática e adaptativa, as características de baixo e alto nível disponíveis nas imagens. A CNN normalmente é composta por dois grupos de trabalho, o primeiro grupo é o responsável pelo processo de extração de características dos dados de entrada (imagens), e o segundo grupo tem a responsabilidade de efetuar a classificação a partir dos dados extraídos do primeiro grupo. O primeiro grupo possui duas camadas principais, sendo elas, a convolução (*convolutional layer*) e o agrupamento (*pooling layer*) e, no último grupo, é tido a camada totalmente conectada (*fully-connected layer*). Estas representações podem ser visualizadas de maneira macro através da Figura 3.

Figura 3 – Arquitetura macro de uma CNN



Fonte – Danilo Assunção, 2022

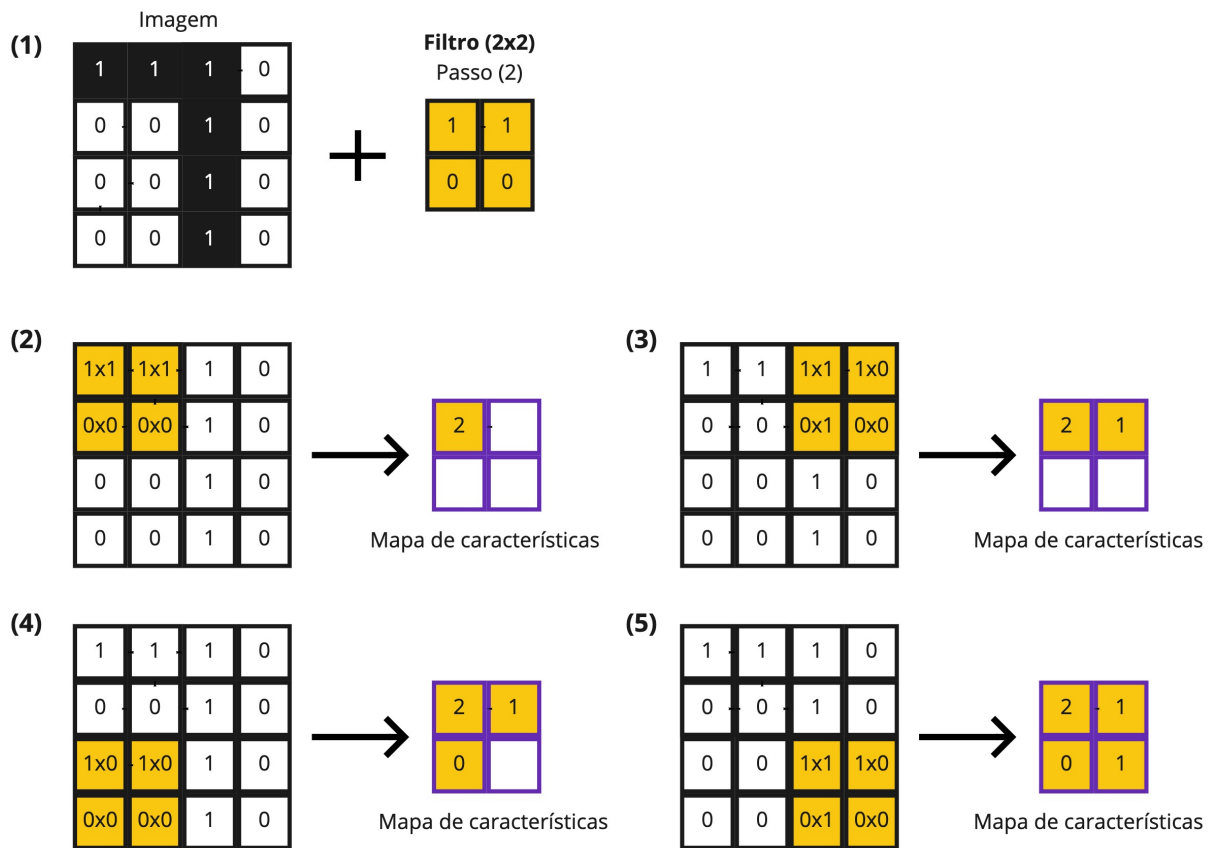
As camadas de convolução e agrupamento executam a extração de características, enquanto a camada totalmente conectada mapeia as características extraídas na saída final, como a classificação.

2.5.1 Camada convolucional

A camada convolucional (*convolutional layer*) é um componente fundamental da arquitetura de uma CNN, tendo como objetivo a extração de características de uma imagem digital, que consiste normalmente em uma combinação de operações matemáticas para se realizar tal tarefa (YAMASHITA *et al.*, 2018).

De acordo com as definições em O'Shea e Nash (2015), a convolução pode ser tratada como um tipo especializado de operação linear, usada para o processo de extração de características em que, um filtro pequeno em dimensionalidade espacial é aplicado em uma matriz de valores (imagem) que se espalha ao longo de toda a profundidade da entrada. Quando uma imagem atinge uma camada convolucional, esta camada *convoluciona* cada filtro na imagem, para assim produzir um mapa de características. Conforme o filtro é navegado pela imagem, o produto escalar é calculado para cada valor do filtro, como é mostrado na Figura 4. A partir disso, na camada convolucional, a rede aprenderá os filtros que serão excitados (ativados) quando veem uma característica específica em uma determinada posição espacial da imagem.

Figura 4 – Processo da convolução em dados de entrada



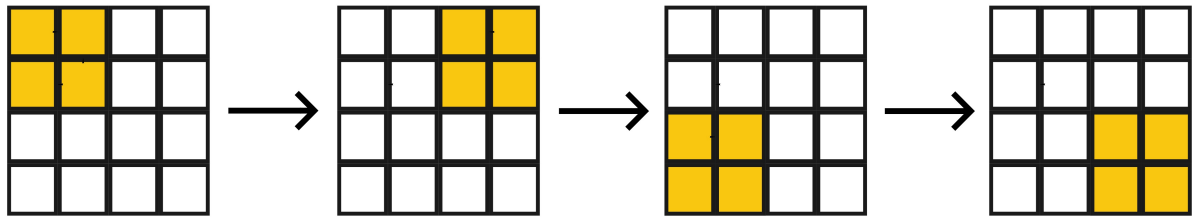
Fonte – Danilo Assunção, 2022

Nas definições em O'Shea e Nash (2015), o filtro de convolução possui alguns hiperparâmetros importantes, sendo os principais, dimensão espacial do filtro, passo (*stride*) e o preenchimento (*padding*).

A dimensão espacial do filtro é definida proporcionando valores que definem o aspecto de dimensão do filtro para convolução, podendo assumir uma dimensão de 3x3, entretanto filtros de 5x5 ou 7x7 também podem ser usados. Este tamanho vai de acordo com a aplicabilidade, ou com o domínio abordado.

O passo especifica o quanto movemos o filtro de convolução em cada etapa. Ou seja, se definíssemos o valor do passo como 1, teríamos um campo receptivo (filtro) sendo envolvido em passos de 1 da esquerda para a direita e de cima para baixo por toda a dimensão espacial da imagem. Alternativamente, definir o passo para um número maior, reduzirá a quantidade de sobreposição e produzirá uma saída de dimensões espaciais menores. Na Figura 5 é demonstrado um exemplo usando um passo de valor 2.

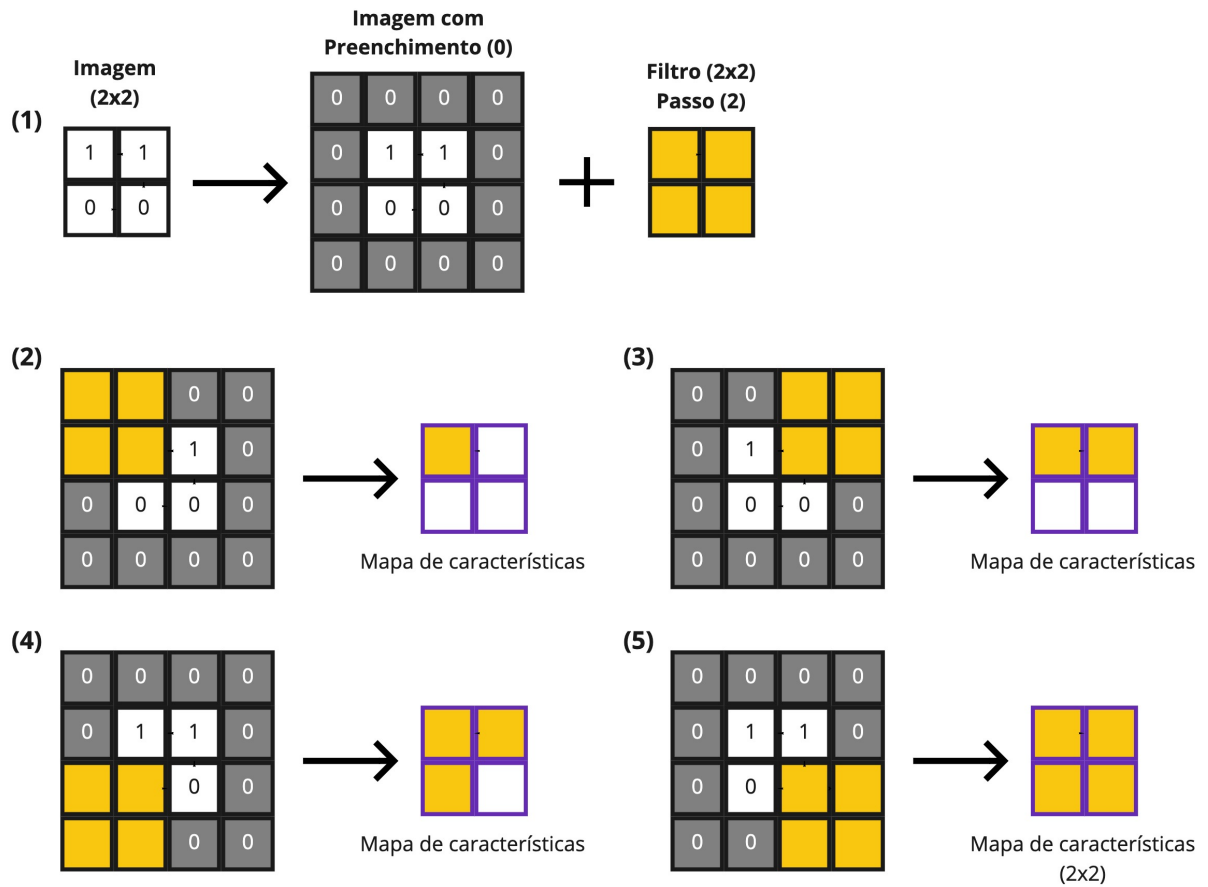
Figura 5 – Exemplo do uso do passo de valor 2 no processo convolucional



Fonte – Danilo Assunção, 2022

Na definição em [Yamashita *et al.* \(2018\)](#), o preenchimento é um método em que linhas e colunas de zeros são adicionadas em cada lado da matriz de valores que representa a imagem, de modo a ajustar o centro de um filtro no elemento mais externo, e mantendo a mesma dimensão no plano através da operação de convolução, sem o preenchimento, cada mapa de características sucessivo ficaria menor após a operação de convolução. Arquiteturas modernas para CNN geralmente empregam preenchimento zero. Na Figura 6 é ilustrada a aplicação de preenchimento de 0 onde é possível observar que a imagem original e o mapa de características gerado possuem a mesma dimensão espacial.

Figura 6 – Exemplo do uso de preenchimento de valor 0 no processo convolucional



Fonte – Danilo Assunção, 2022

2.5.2 Função de ativação não linear

As saídas de uma operação linear, como a convolução, passam por uma função de ativação não linear. A função de ativação mais comum usada atualmente é a unidade linear retificada (ReLU), devido a facilidade em se computar suas derivadas melhorando o desempenho da rede em relação a este procedimento (ALBAWI; MOHAMMED; AL-ZAWI, 2017). A ReLU é expressada pela Equação 5:

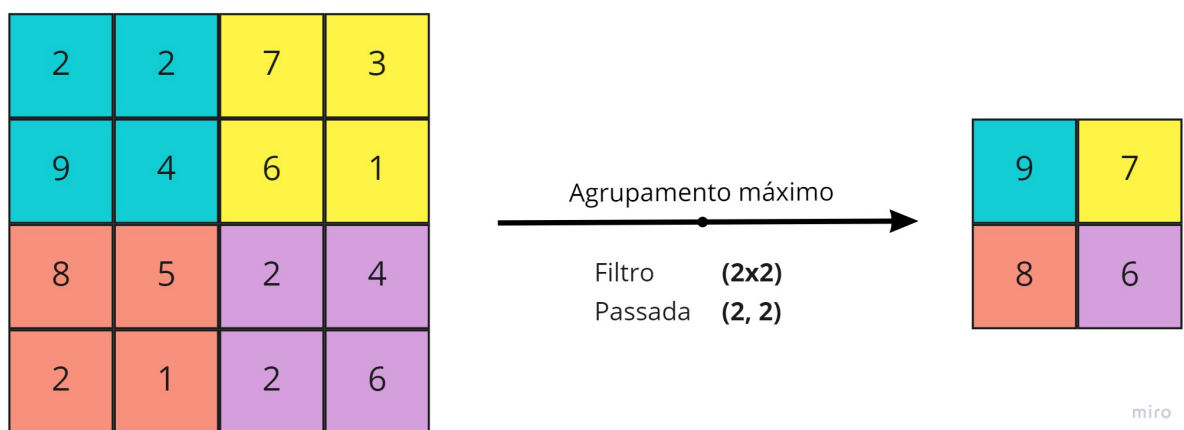
$$ReLU(x) = \max(0, x) \tag{5}$$

2.5.3 Camada de agrupamento

O objetivo da camada de agrupamento (*pooling layer*) é tornar a representação de características conjuntas mais utilizável, que preserve informações importantes enquanto descarta detalhes irrelevantes (YU *et al.*, 2014). Ou seja, a camada de agrupamento visa a reduzir gradualmente a dimensionalidade da representação e, assim, reduzir ainda mais o número de parâmetros e complexidade computacional do modelo (O'SHEA; NASH, 2015). A operação oferecida pelo agrupamento é tipicamente conhecida como *downsampling*. É importante notar que não há nenhum parâmetro treinável nas camadas de agrupamento, enquanto o tamanho do filtro, a passada (*stride*) e o preenchimento (*padding*) são também hiperparâmetros em operações de agrupamento, semelhantes às operações de convolução (YAMASHITA *et al.*, 2018). Existem algumas funções de agrupamento presentes na literatura, abaixo será explorado o agrupamento máximo e o agrupamento médio.

Agrupamento máximo (*max pooling*) é a forma mais popular de operação de agrupamento, que extrai um mapa de características condensado através do mapa de características recebido da saída da camada convolucional, obtendo apenas o valor máximo contido no mapa de características recebido e descartando os demais valores (YAMASHITA *et al.*, 2018). Este método pode ser observado pela Figura 7.

Figura 7 – Agrupamento máximo

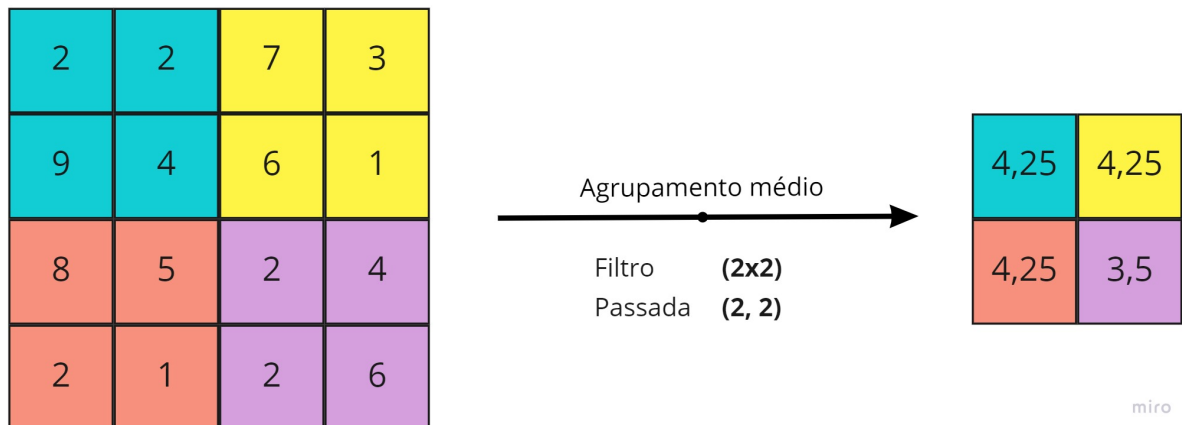


Fonte – Danilo Assunção, 2022

O agrupamento médio (*average pooling*) extrai um mapa de características condensado do mapa de características recebido a partir do cálculo da média destes elementos. Assim, enquanto o agrupamento máximo fornece a característica mais proeminente em um

mapa de características condensado, o agrupamento médio calcula a média dos elementos presentes na região do mapa de características coberto pelo filtro. Este método é ilustrado na Figura 8.

Figura 8 – Agrupamento médio



Fonte – Danilo Assunção, 2022

2.5.4 Camada totalmente conectada

Os mapas de características de saída da convolução final ou camada de agrupamento, são normalmente achatados (*flattened*), ou seja, transformados em uma matriz unidimensional (ou vetor) e conectados a uma ou mais camadas totalmente conectadas, também conhecidas como camadas densas, em que todas as entradas são conectadas a todas as saídas por um peso que se ajusta em seu procedimento de aprendizado (YAMASHITA *et al.*, 2018). A camada final totalmente conectada normalmente possui o mesmo número de nós de saída que o número de classes relacionadas ao domínio trabalhado no modelo (YAMASHITA *et al.*, 2018).

2.6 Técnica de transferência de aprendizado

A transferência de aprendizado visa a extrair o conhecimento de uma ou mais tarefas de origem e aplicar o conhecimento a uma tarefa de destino (PAN; YANG, 2009). Esta técnica funciona treinando uma rede em um elevado conjunto de dados, como *ImageNet* que é um grande conjunto de dados visuais projetado para uso em pesquisa de software de reconhecimento de objetos visuais (RUSSAKOVSKY *et al.*, 2015), para então usar tais

pesos como os pesos iniciais em uma nova tarefa de classificação. Segundo [Shao, Zhu e Li \(2014\)](#), a eficiência desta técnica é notada pois muitos conjuntos de dados de imagem compartilham características espaciais de baixo nível, que são melhor compreendidas a partir de um grande volume de dados. A transferência de aprendizado possui alguns métodos conhecidos na literatura, entretanto, será dada ênfase a dois métodos contextualizados nesta técnica, sendo tais métodos, a extração de características e o ajuste fino.

Baseado na descrição em [Li e Hoiem \(2017\)](#), o método de extração de características (*feature extraction*) usa uma CNN pré-treinada para extrair características de uma imagem. Classificadores treinados nessas características podem alcançar resultados competitivos, às vezes superando características projetadas manualmente (*handcrafted*). Este método não modifica a rede original e permite que novas tarefas se beneficiem de características complexas aprendidas em tarefas anteriores. No entanto, esses recursos não são especializados para a nova tarefa e muitas vezes podem ser aprimorados através do método ajuste fino.

O método ajuste fino (*fine-tuning*) modifica os parâmetros de uma CNN pré-treinada existente para treinar em uma nova tarefa. Para isto, uma pequena taxa de aprendizado é frequentemente usada e, às vezes, parte da rede é congelada para evitar sobreajuste. Usando hiperparâmetros apropriados para treinamento, o modelo resultante geralmente supera o método de extração de características ou o aprendizado de uma rede inicializada aleatoriamente. O ajuste fino torna o processo de obtenção das características da imagem mais discriminativo para a nova tarefa, e a baixa taxa de aprendizado é um mecanismo indireto para preservar parte da estrutura representacional aprendida nas tarefas originais.

2.7 Técnica de aumento de dados

Conforme descrito em [Shorten e Khoshgoftaar \(2019\)](#), a técnica de aumento de dados (*data augmentation*) abrange um conjunto de métodos que aumentam o tamanho dos conjuntos de dados de treinamento, nessa conformidade, melhores modelos de aprendizado profundo podem ser construídos com eles devido a sua capacidade em minimizar o sobreajuste. A seguir são descritos alguns métodos incluídos nesta técnica que são comumente utilizados:

1. **Inversão:** este método consiste em inverter o eixo horizontal ou vertical de uma imagem, em que a inversão do eixo horizontal costuma ser mais frequente do que a

inversão do eixo vertical. Esta técnica é uma das mais simples de se implementar, e provou ser útil em determinados conjuntos de dados de imagens.

2. **Isolamento de canais de cores:** os dados de uma imagem digital são geralmente codificados como uma matriz tridimensional (altura \times largura \times canais de cor). Este método inclui no isolamento de um simples canal de cor, tal como R, G ou B, sendo que a imagem pode ter sua representação de cores rapidamente convertida ao isolar um destes canais tornando os canais restantes preenchidos com valores de 0. Adicionalmente, os valores nos canais de cores da imagem (RGB) podem ser manipulados com simples operações de matriz.
3. **Transformação fotométrica:** este método consiste em percorrer as imagens e diminuir ou aumentar os valores dos *pixels* para uma determinada constante. Outra transformação consiste em restringir os valores dos *pixels* a um determinado valor mínimo ou máximo. A representação intrínseca da cor em imagens digitais se presta a muitas estratégias de ampliação. Nesta técnica estão incluídos os métodos empregados para ajustes de iluminação da imagem ou na conversão da imagem para uma escala de cinza.
4. **Corte:** cortar imagens é uma prática de processamento, em que áreas relevantes da imagem são destacadas e redimensionadas. Por exemplo, uma imagem de dimensão 256x256 em que, após o corte, ficou com dimensão 224x224. Dependendo do limite de redução escolhido para o corte, pode ser uma transformação que pode ou não preservar o tamanho original da imagem.
5. **Rotação:** é a realização de rotação da imagem para a direita ou esquerda, em um eixo compreendido entre 1° e 359°. A segurança deste método é fortemente determinada pelo grau de rotação, sendo que, à medida que o grau de rotação aumenta, a classe que representa os dados pode não mais ser preservada após aplicada tal transformação.
6. **Deslocamento:** é um método de deslocação de imagens para a esquerda, direita, cima ou baixo, podendo ser uma transformação muito útil para evitar um viés no modelo e preservando as dimensões da imagem. Por exemplo, se todas as imagens em um conjunto de dados estiverem centralizadas, o que é comum em conjuntos de dados de reconhecimento de rosto, exigiria que o modelo também fosse testado em imagens perfeitamente centradas. Como a imagem original foi deslocada para um direção, o espaço restante pode ser preenchido com um valor constante, no caso de

um modelo na escala RGB, poderia ser aplicada a cor preta (0, 0, 0) ou a cor branca (255, 255, 255).

2.8 Conclusão

Neste capítulo uma série de conceitos importantes foram apresentados que estão amplamente relacionados a este trabalho de pesquisa, que envolve a aplicação de Redes Neurais Convolucionais (CNNs) na resolução de problemas de reconhecimento de espécies. Contudo, as explicações para os conceitos de aprendizado de máquina e suas ramificações de mais baixo nível, como aprendizado profundo e algoritmos de CNN foram todas especificadas nesta seção, a fim de permitir um maior entendimento das definições utilizadas nos capítulos a seguir.

3 Revisão sistemática

A revisão sistemática possibilita uma forma de análise e instrução a partir das informações dispostas pelos artigos disponíveis para uma pesquisa em específico, assunto de tópico ou tema de interesse, sendo que deve ser utilizada uma sistemática de confiança, auditável e integra (HOLGADO *et al.*, 2020). O desenvolvimento da revisão sistemática se baseia em um protocolo devidamente definido para alcançar resultados válidos e detalhados, trazendo assim a possibilidade de análise e contradição destes resultados a partir do uso do protocolo determinado.

3.1 *Objetivos da revisão*

Na área de visão computacional há a aplicação de técnicas com objetivo de identificação de insetos, cuja técnica de processamento, baseado em imagens e algoritmos, acabam por reconhecer os padrões presentes nas imagens e, assim, classificar automaticamente as espécies de insetos. Tal processamento vem modificando o tradicional modelo manual e descritivo de características morfológicas já fornecido por estudos taxonômicos para a identificação destes insetos. Contudo, apenas especialistas como taxonomistas e técnicos qualificados podem identificar tais insetos com precisão, pois isso requer conhecimento especializado e obtido por meio de uma vasta experiência (LIM; KIM; KIM, 2017).

O aprendizado profundo tem sido amplamente utilizado em processamento de imagens, visão computacional e reconhecimento de padrões. Diferente dos métodos tradicionais de aprendizado de máquina, o modelo de aprendizado profundo pode aprender características automaticamente a partir de uma grande quantidade de imagens, e não requer um especialista com domínio na construção do extrator de características (LIU *et al.*, 2020b).

As características extraídas das imagens são os fatores que mais influenciam na acurácia do classificador. Embora existam várias técnicas e ferramentas para extração de características automaticamente, ainda não se encontram trabalhos na literatura que avaliem estas técnicas de aprendizado profundo e nem ferramentas que as entreguem de forma sistemática.

Portanto, o objetivo desta revisão sistemática é apresentar e discutir o estado da arte das técnicas de reconhecimento automático de espécies por meio de suas imagens e morfologias diversas, empregando técnicas de aprendizado profundo. Visando tal objetivo, foram analisados os artigos com foco na aplicação de Redes Neurais Convolucionais (CNN).

3.2 *Protocolo*

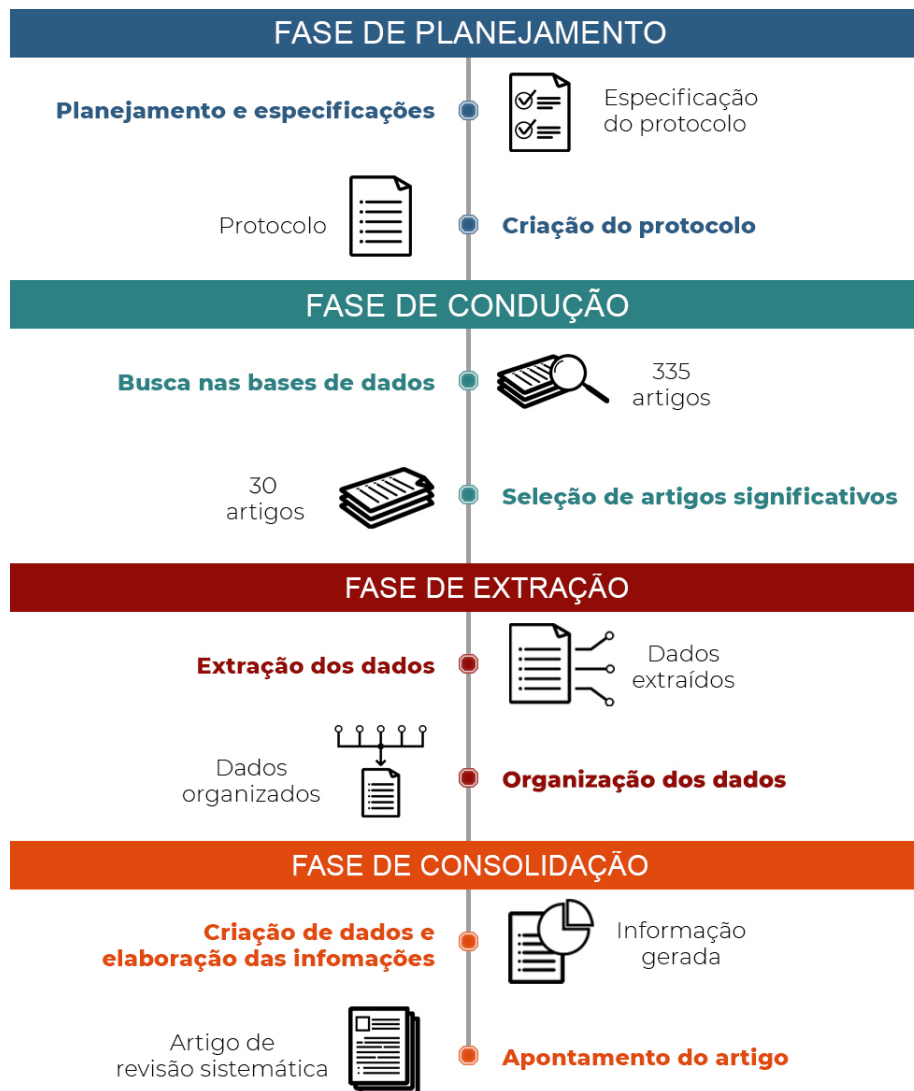
Foram definidas quatro fases para a realização desta revisão sistemática, sendo elas:

1. Planejamento
2. Condução
3. Extração
4. Consolidação

O planejamento consiste na criação do protocolo no qual as diretrizes da pesquisa serão baseadas. A condução é a execução da pesquisa e seleção de trabalhos de interesse, de acordo com os critérios de inclusão e exclusão definidos no protocolo. A extração é a fase na qual os artigos que foram selecionados são estudados e os dados importantes são separados. Em suma, temos a consolidação, na qual os dados são cruzados para gerar conhecimento e, assim, o artigo de revisão sistemática é escrito.

Estes processos estão representados na Figura 9, na qual cada etapa do processo está identificada com uma cor diferente.

Figura 9 – Diagrama do processo de revisão sistemática



Fonte – Danilo Assunção, 2021

Na etapa de planejamento foi levantada junto ao protocolo a seguinte questão: “Quais os métodos, adeptos ao estado da arte, para a identificação de espécies de insetos usando-se uma Rede Neural Convolutacional por meio de imagens digitais?”. A conclusão desta pergunta veio a partir de bases de dados junto às seguintes *strings* de busca:

- **IEEE Xplore** - (“convolutional neural network” OR “cnn” OR “convolution”) AND “feature extraction” AND (“insect” OR “pest”) AND (“image” OR “image segmentation” OR “image recognition”)
- **SCOPUS** - (“convolutional neural network” OR “cnn” OR “convolution”) AND “feature extraction” AND (“insect” OR “pest”) AND (“image” OR “image segmentation” OR “image recognition”) AND (EXCLUDE (DOCTYPE, “cp”) OR

EXCLUDE (DOCTYPE, “re”) OR EXCLUDE (DOCTYPE, “bk”) OR EXCLUDE (DOCTYPE, “ch”))

A palavra chave *pest* é incluída nas *strings* de busca por se tratar de um objeto de estudo muito recorrente nos meios de reconhecimento de espécies através de imagens, utilizando de técnicas similares às que são usadas para reconhecimento de insetos por meio de métodos utilizando CNN.

Este trabalho foi conduzido de dezembro de 2020 até fevereiro de 2021. Inicialmente 335 trabalhos foram encontrados nas bases de dados especificadas acima, dos quais 302 estavam na base de dados SCOPUS e 33 na base de dados IEEE Xplore.

Após aplicar os critérios de inclusão e exclusão nos trabalhos coletados, 30 artigos foram selecionados e analisados por completo, sendo 27 da base SCOPUS e 3 da base IEEE Xplore.

Os critérios utilizados para incluir ou excluir um trabalho nesta revisão foram:

- Inclusão
 - Artigos que utilizam CNNs como ferramenta para discriminação de espécies por meio de imagens;
 - Artigos que contêm técnicas computacionais de segmentação e extração de características de imagens de diversas espécies biológicas;
 - Artigos devem ter data de publicação maior ou igual a 2015.
- Exclusão
 - Artigos que não são relacionados com conceitos envolvendo técnicas de segmentação e extração de características por meio de imagens;
 - Artigos que não utilizam CNNs para classificação automática de espécies por meio de imagens;
 - Artigos que não foram escritos nos idiomas inglês ou português;
 - Artigos com data de publicação menor que 2015;
 - Documentos que não estejam categorizados como artigo em seu tipo de documento.

Com a finalidade de poder encontrar possíveis lacunas nas pesquisas utilizadas, seguidamente da extração de dados, foi exercida uma averiguação sobre os principais obstáculos e propensões praticadas.

3.3 Resultado da revisão sistemática

Os 30 trabalhos selecionados foram estudados e os seguintes atributos foram extraídos: ano de publicação, técnicas de pré-processamento de imagens, técnicas de transferência de aprendizado, técnicas de aumento de dados, modelo de arquitetura da CNN e métodos de classificação. Um breve sumário dos trabalhos com os respectivos dados extraídos encontram-se no Tabela 1.

Tabela 1 – Sumário de todos os artigos analisados

Artigo	Ano de publicação	País de afiliação dos autores	Téc. de processamento de imagens	Téc. de transferência de aprendizado	Téc. de aumento de dados	Modelo de arquitetura da CNN	Modelo do classificador
(LIU <i>et al.</i> , 2020b)	2020	Estados Unidos	Não aplicado	Ajuste fino	<i>Skewing</i> , Distorção elástica, Rotação	LeNet-5, AlexNet, VGGNet-16, VGGNet-19, ResNet-50, Inception v3, InceptionResNet v2	Totalmente conectada
(LE <i>et al.</i> , 2020)	2020	França	Redimensionamento	Ajuste fino	Constante para cada canal de cor, Divisão dos canais da imagem	EB-Net	Totalmente conectada
(ZHANG; CHEN,)	2020	China	Não especificado	Não especificado	Não especificado	DenseNet-121	Não especificado
(LIU <i>et al.</i> , 2020a)	2020	China	Não aplicado	Não aplicado	Alto brilho, Baixo brilho, Alto contraste, Baixo contraste, Alta nitidez, Baixa nitidez, Rotação, Simetria vertical, Simetria horizontal, Ruído gaussiano, <i>PCA Jittering</i>	DICNN	Totalmente conectada
(LU; TUNG; KUO, 2020)	2020	Taiwan	Redimensionamento	Ajuste fino	Inversão horizontal, Inversão vertical, Deslocamento horizontal, Deslocamento vertical, Rotação, <i>Shearing</i> , Ampliação	VGGNet-16	Totalmente conectada

Continua na próxima página

Tabela 1 – continuação da página anterior

Artigo	Ano de publicação	País de afiliação dos autores	Téc. de processamento de imagens	Téc. de transferência de aprendizado	Téc. de aumento de dados	Modelo de arquitetura da CNN	Modelo do classificador
(BANAN; NA-SIRI; TAHERI-GARAVAND, 2020)	2020	Irã	Redimensionamento	Ajuste fino	Rotação, Deslocamento vertical, Deslocamento horizontal	VGGNet-16	Totalmente conectada
(TIWARI, 2020)	2020	Índia	Não especificado	Não aplicado	Não aplicado	Indefinido	Totalmente conectada
(LI <i>et al.</i> , 2020)	2020	China	Recorte, Redimensionamento	Não aplicado	Rotação, Recorte	VGGNet-16, Inception v3	Totalmente conectada
(CHEN <i>et al.</i> , 2020)	2020	China	Redimensionamento	Ajuste fino	Rotação, Borrado, Oclusão	ResNeXt-101	Totalmente conectada
(HONG <i>et al.</i> , 2020)	2020	China	Redimensionamento, Recorte	Ajuste fino	Filtro gaussiano, Aprimoramento de contraste	AlexNet	Totalmente conectada
(RAUF <i>et al.</i> , 2019)	2019	Paquistão, Estados Unidos	Redimensionamento, Fundo transparente	Não aplicado	Não aplicado	32-Layer CNN	Totalmente conectada
(GÓMEZ-RÍOS <i>et al.</i> , 2019)	2019	Espanha, Dinamarca	Não especificado	Ajuste fino	Não especificado	Inception v3, ResNet-50, ResNet-152, DenseNet-121, DenseNet-161	Totalmente conectada
(CIBUK <i>et al.</i> , 2019)	2019	Turquia, Estados Unidos	Não aplicado	Extração de características	Não aplicado	VGGNet-16, AlexNet	SVM (RBF)
(LI <i>et al.</i> , 2019)	2019	China	Redimensionamento	Não aplicado	Multi-scale, Rotação	ResNet-50	Totalmente conectada

Continua na próxima página

Tabela 1 – continuação da página anterior

Artigo	Ano de publicação	País de afiliação dos autores	Téc. de pré-processamento de imagens	Téc. de transferência de aprendizado	Téc. de aumento de dados	Modelo de arquitetura da CNN	Modelo do classificador
(KHALIFA <i>et al.</i> , 2019)	2019	Egito, Luxemburgo	Não especificado	Não aplicado	Reflexão, Ampliação, Ruído gaussiano, Ruído de poisson, Efeitos de granulação, Ruído de manchas	Indefinido	Totalmente conectada
(HSIANG <i>et al.</i> , 2019)	2019	Suécia, Reino Unido	Redimensionamento	Ajuste fino	Não especificado	VGGNet-16, DenseNet-121, Inception v3	Totalmente conectada
(REN; LIU; WU, 2019)	2019	Japão	Redimensionamento, Recorte	Não aplicado	Redimensionamento, Recorte, Normalização de média e desvio padrão	FR-ResNet	Totalmente conectada
(VALAN <i>et al.</i> , 2019)	2019	Suécia	Redimensionamento, Recorte	Extração de características	Não aplicado	VGGNet-16	SVM (Linear)
(GYIRES-TÓTH <i>et al.</i> , 2019)	2019	Hungria	Recorte, Redimensionamento	Ajuste fino	Não aplicado	AlexNet, Inception V3	Totalmente conectada
(THENMOZHI; REDDY, 2019)	2019	Índia	Escala de cinza, Detecção de bordas de canny, Recorte, Redimensionamento	Ajuste fino	Reflexão, Escala, Rotação, Translation	VGGNet-16, VGGNet-19	Totalmente conectada
(TETILA <i>et al.</i> , 2019)	2019	Brasil	Não especificado	Ajuste fino, Extração de características	Rotação, Escala, <i>Scrolling</i> , Ampliação	DenseNet-201, InceptionResNet v2, ResNet-50	Totalmente conectada
(TAN <i>et al.</i> , 2018)	2018	Malásia	Redimensionamento, Preenchimento	Não aplicado	Não aplicado	D-Leaf	Totalmente conectada

Continua na próxima página

Tabela 1 – continuação da página anterior

Artigo	Ano de publicação	País de afiliação dos autores	Téc. de processamento de imagens	Téc. de transferência de aprendizado	Téc. de aumento de dados	Modelo de arquitetura da CNN	Modelo do classificador
(ZHU; ZHU; REN, 2018)	2018	China	Não especificado	Não aplicado	Não aplicado	Inception v2	Totalmente conectada
(ZHU <i>et al.</i> , 2018)	2018	China	Redimensionamento	Não especificado	Não aplicado	Indefinido	SVM (Linear)
(VILLON <i>et al.</i> , 2018)	2018	França	Não especificado	Não aplicado	Não aplicado	GoogLeNet	Totalmente conectada
(LEE <i>et al.</i> , 2017)	2017	Malásia, Reino Unido	Redimensionamento	Ajuste fino	Rotação	Indefinido	Totalmente conectada
(ZIELIŃSKI <i>et al.</i> , 2017)	2017	Polónia	Não especificado	Extração de características	Não aplicado	AlexNet, VGGNet-M, VGGNet-VD	SVM (Linear, RBF e Polinomial), Floresta aleatória
(WANG <i>et al.</i> , 2017)	2017	China	Não especificado	Não especificado	Não aplicado	LeNet-5, AlexNet	Totalmente conectada
(LIM; KIM; KIM, 2017)	2017	Coreia do Sul	Redimensionamento, Recorte	Não especificado	Rotação, Ampliação	AlexNet	Totalmente conectada
(SUN <i>et al.</i> , 2017)	2017	China	Não especificado	Não aplicado	Não aplicado	ResNet-26	Totalmente conectada

Fonte - Danilo Assunção, 2021

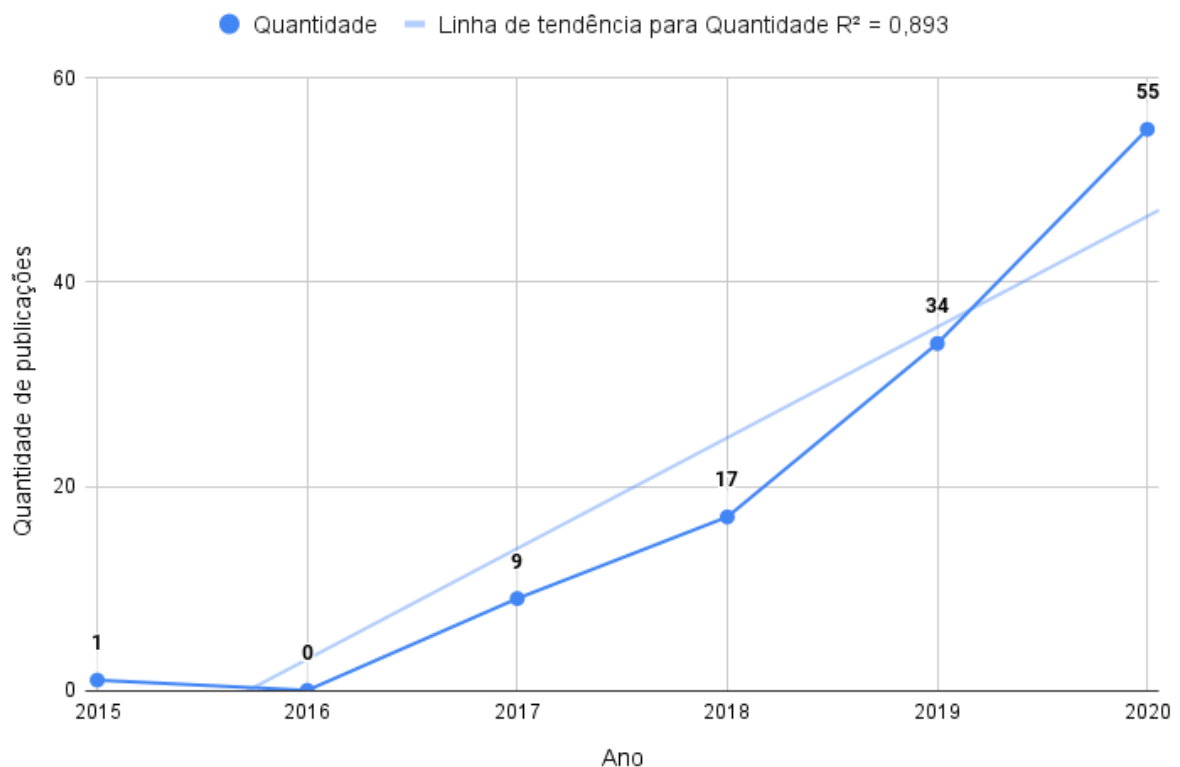
3.3.1 Ano de publicação

Nesta revisão sistemática foi coletado o ano de publicação dos artigos, dessa maneira, fez-se uma análise de tendência referente aos assuntos que abordam técnicas de reconhecimento de espécies utilizando uma CNN, em domínios que envolvem assuntos relacionados ao reconhecimento de espécies.

Visando a uma melhor compreensão sobre a quantidade de publicações com base em assuntos que envolvem reconhecimento automático de espécies utilizando técnicas de aprendizado profundo, foi elaborado um gráfico com todos os artigos encontrados na base de dados pela consulta feita nas bases do SCOPUS e IEEE Xplore.

Na Figura 10 é apresentado uma linha de tendência aos assuntos envolvendo os artigos encontrados pela *string* de busca que possuem as palavras chaves relacionadas com, aprendizado profundo, CNNs e reconhecimento de espécies.

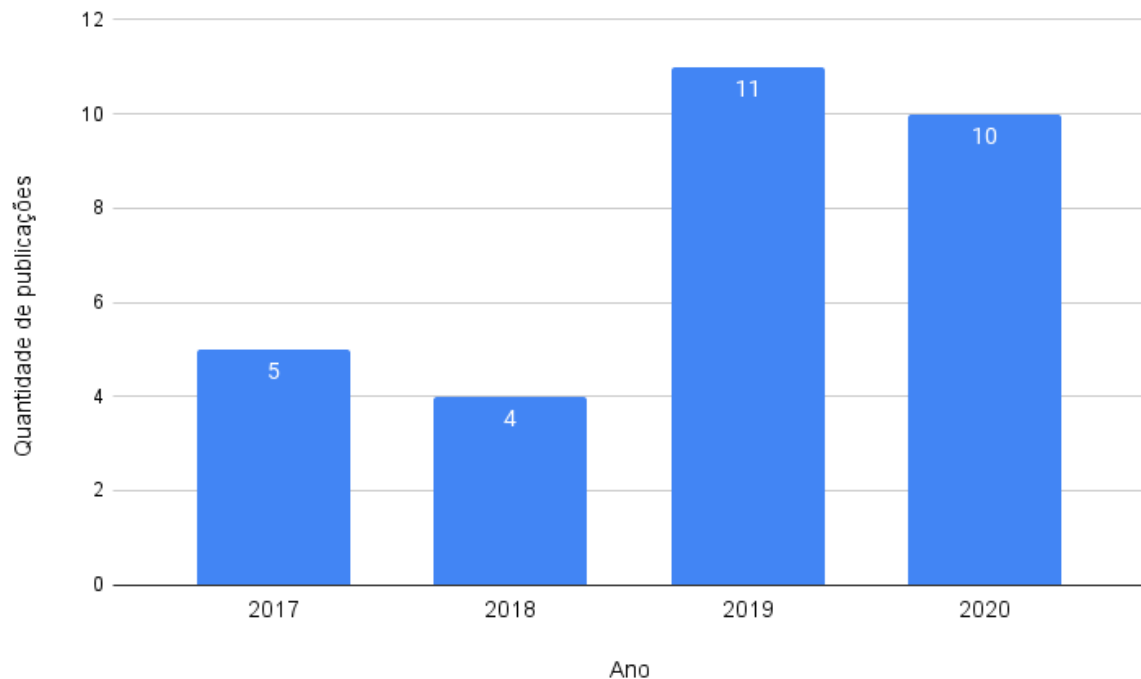
Figura 10 – Ano de publicação dos artigos com base em palavras chaves



Fonte – Danilo Assunção, 2021

Na Figura 11 é apresentada uma relação de quantidade versus ano, para o total de artigos que foram filtrados e selecionados a partir do conjunto obtido nas consultas executadas nas bases do SCOPUS e IEEE Xplore para esta revisão sistemática.

Figura 11 – Ano de publicação dos artigos selecionados



Fonte – Danilo Assunção, 2021

Nota-se que a maior quantidade dos artigos selecionados foi publicada em 2019, representando um total de 36,67% (11). Já em 2020, é tido 33,33% (10) dos artigos. Em 2018 a quantidade de artigos foi de 13,33% (4) e em 2017, 16,67% (5).

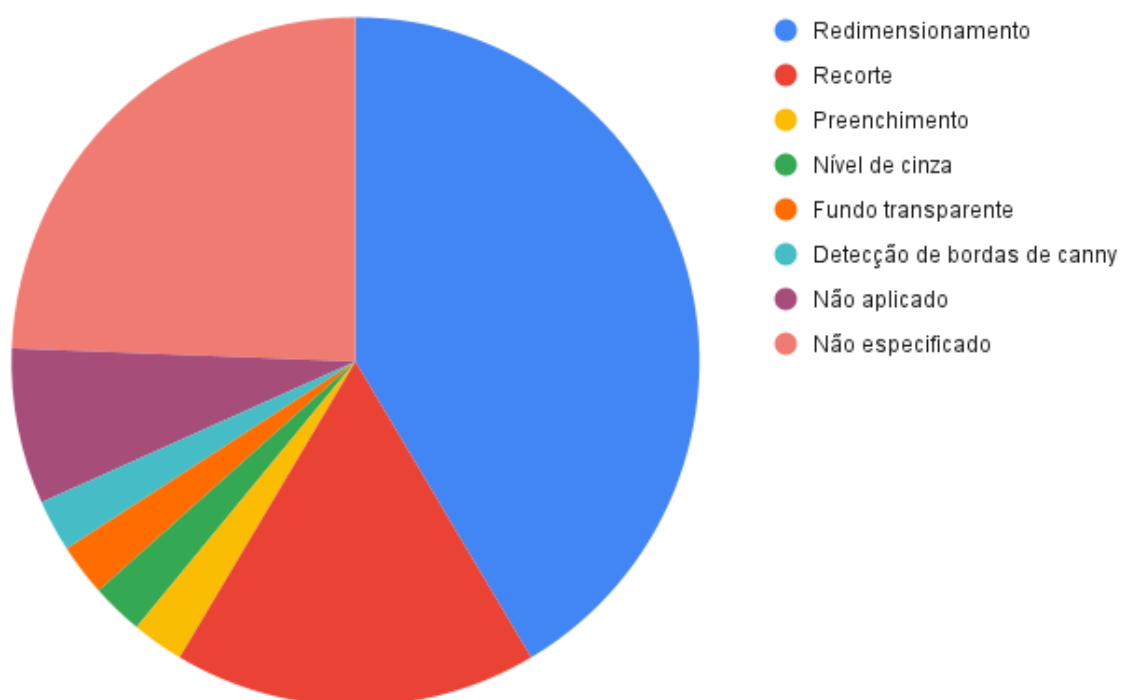
É possível perceber que, desde 2017 até 2020, houve um aumento significativo de artigos de interesse publicados com relação à aplicação de CNNs que almejam a resolução de problemas envolvendo reconhecimento de espécies em imagens digitais. Essa representatividade indica a veracidade de uma tendência de alta, com relação à utilização de técnicas envolvendo CNNs neste tipo de aplicação (LIU *et al.*, 2020b).

Os artigos de interesse obtidos possuem data de publicação a partir do ano de 2017, os artigos de 2015 a 2016 não se mostraram muito relevantes para os assuntos relacionados à esta revisão sistemática, sendo levado em consideração os critérios de exclusão definidos.

3.3.2 Técnicas de pré-processamento de imagens

Nos métodos abordados nos artigos selecionados é possível identificar a utilização de técnicas de pré-processamento de imagens em alguns cenários distintos como é apresentado na Figura 12.

Figura 12 – Técnicas de pré-processamento de imagens aplicadas



Fonte – Danilo Assunção, 2021

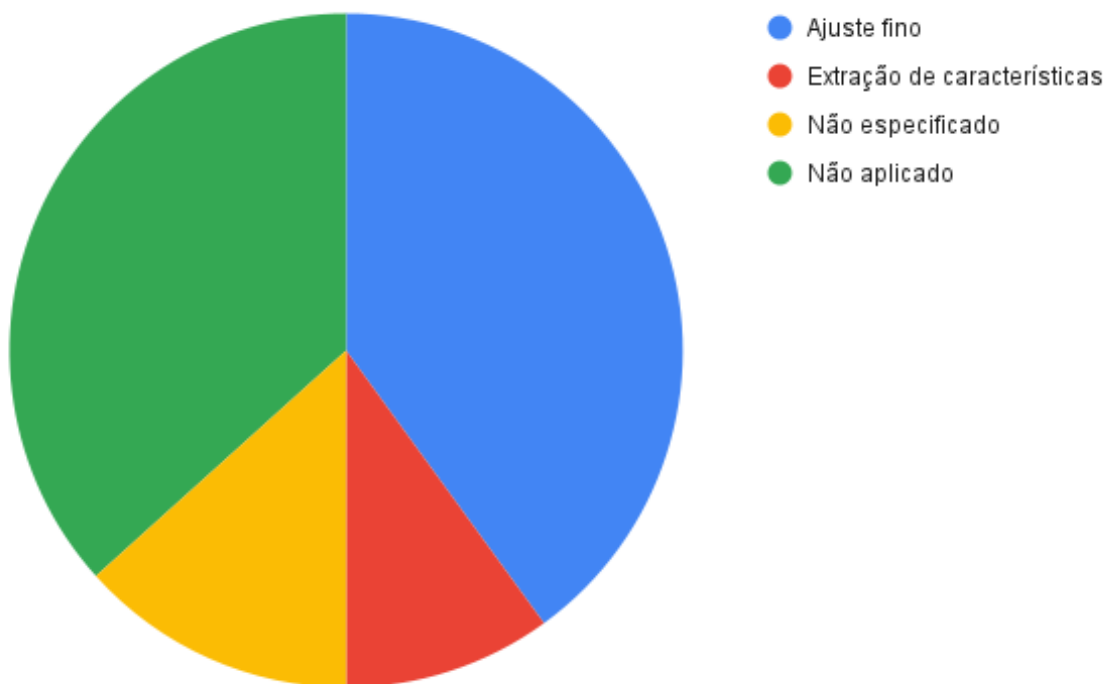
O papel da técnica de pré-processamento, em cenários de classificação por meio de imagens, é um processo importante, pois ele permite destacar as características importantes que podem vir a melhorar o desempenho do classificador (THENMOZHI; REDDY, 2019). A técnica de Redimensionamento lidera com 41,46% de uso com relação às outras estratégias abordadas, e isso se deve aos fatores que auxiliam no aumento do desempenho do classificador, como a quantidade reduzida de parâmetros dado que a imagem é normalmente reduzida. Em segundo, com 17,07%, o Recorte que pode vir a ser utilizado para destacar a região de interesse da imagem e reduzindo também o tamanho da imagem (THENMOZHI; REDDY, 2019). Preenchimento, Nível de cinza, Fundo transparente e Deteção de bordas de canny foram algumas das abordagens utilizadas, e ocupam apenas 2,44% de uso com relação aos outros métodos. Já os 31,71% restantes, são artigos que

não aplicam ou não especificam diretamente o uso de técnicas de pré-processamento de imagens, e isto pode ser ocasionado pelo fato do conjunto de dados utilizado já estar previamente pré-processado em sua disponibilização, ou de não haver a necessidade direta de uma preparação de tais dados.

3.3.3 Técnica de transferência de aprendizado

Dois métodos foram comumente utilizados como técnicas de transferência de aprendizado, sendo elas, o método ajuste fino e o método extração de características, já em outros trabalhos o método não foi aplicado ou não foi especificado algo relacionado a isso. A Figura 13 ilustra os dados coletados por meio dos artigos analisados que utilizam de técnicas de transferência de aprendizado, na resolução de cenários que abordam o uso de CNNs.

Figura 13 – Técnicas de Transferência de aprendizado



Fonte – Danilo Assunção, 2021

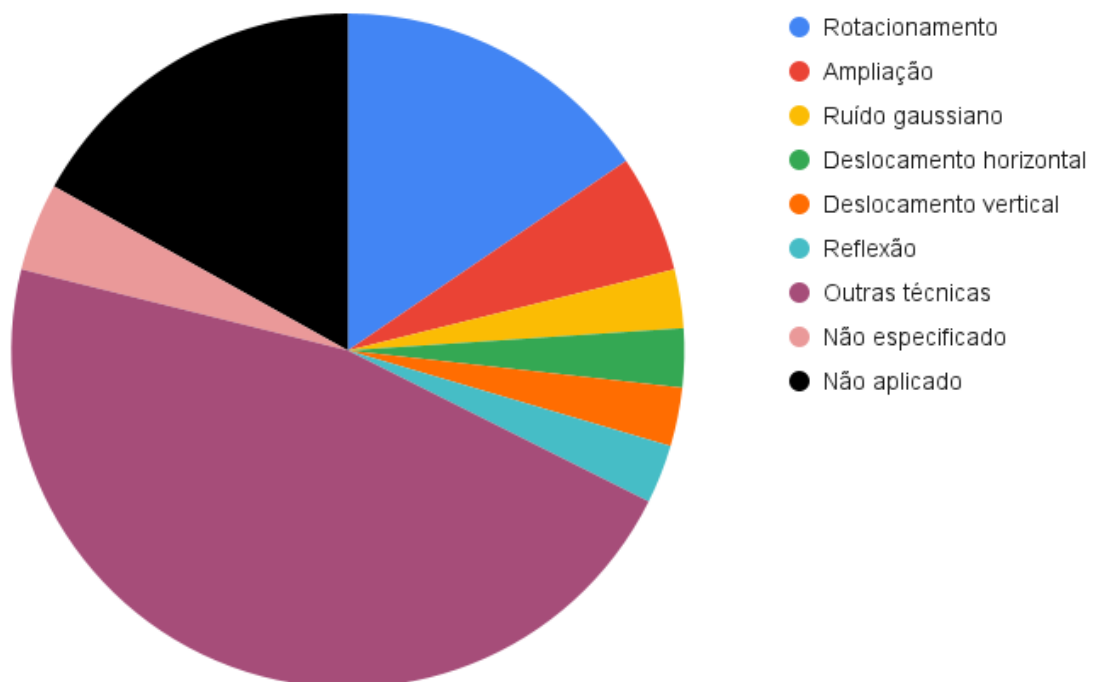
O ajuste fino foi o método mais utilizado com 40% (12) de uso em relação a todas as outras técnicas encontradas nos artigos analisados. Isso revela os aspectos positivos ao se utilizar o método de ajuste fino com CNN pré-treinadas para novas tarefas (LI; HOIEM,

2017), dado que isso auxiliou em diversos trabalhos que compõem este percentual de utilização. O uso do método de extração de características teve uso em apenas 10% (3) dos trabalhos analisados, e este baixo uso também pode implicar que o uso do ajuste fino é normalmente mais comum dado que o ajuste fino pode apresentar dependendo do domínio resultados melhores (LI; HOIEM, 2017). Já nos demais trabalhos analisados, um total de 36,67% (11) não utilizaram esta técnica e 13,33% (4) não especificaram diretamente se houve ou não o uso da técnica de transferência de aprendizado.

3.3.4 Técnicas de aumento de dados

O método de aumento de dados permite aumentar artificialmente o tamanho do conjunto de treinamento. O aumento do conjunto de treinamento é feito ao aplicar várias distorções às imagens originais, como zoom, virando-as horizontalmente ou verticalmente, girando-as, deslocando-as e dentre outros (GÓMEZ-RÍOS *et al.*, 2019). Na Figura 14 são apresentados graficamente os métodos utilizados para o aumento de dados nos artigos abordados.

Figura 14 – Métodos para o aumento de dados

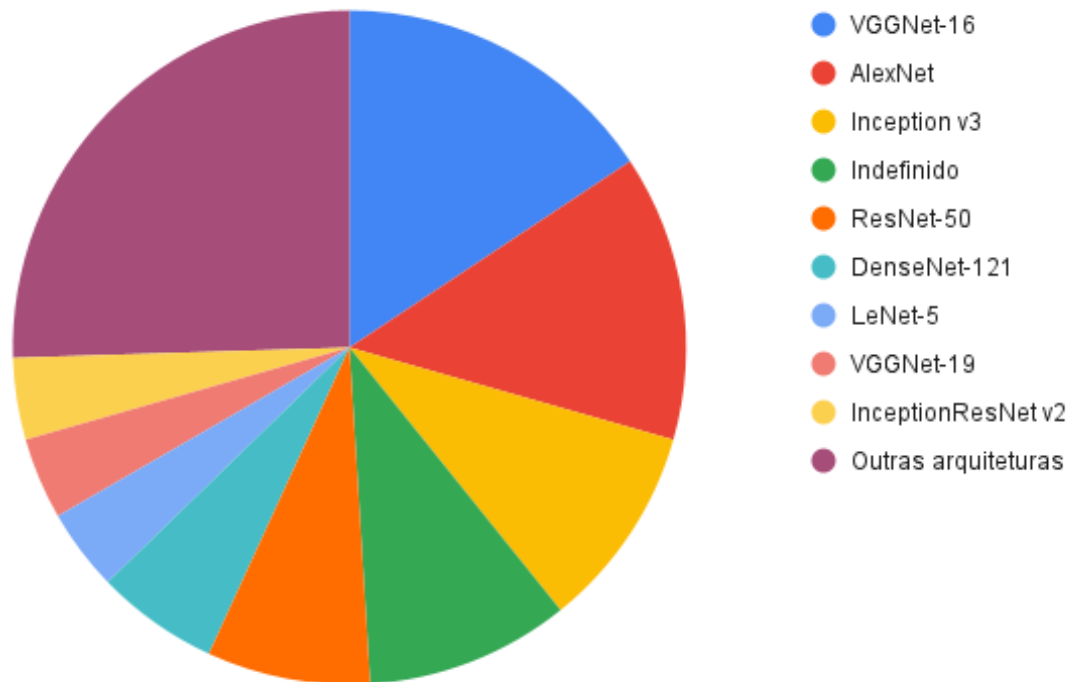


As técnicas de Aumento de dados utilizando Rotacionamento são bem prioritárias quando este mecanismo é aplicado, com um total de uso de 15,49% (11) com relação aos outros meios. Ampliação, sendo a terceira metodologia mais aplicada com 5,63% (4). Ruído gaussiano, Deslocamento horizontal, Deslocamento vertical e Reflexão representando 2,82% (2) nos demais cenários. As Outras técnicas agrupadas são métodos que foram vistos sendo utilizados somente uma única vez nos artigos selecionados e o conjunto não especificado são aos artigos que não esclarecem de maneira explícita o tipo de técnica de Aumento de dados utilizada. Entretanto, a maioria dos artigos não aplicou técnicas de Aumento de dados, com uma representação de 16,90% (12) de todos os exemplares analisados, isto pode ser ocasionado pelo fato dos conjuntos de dados utilizados pelos autores já serem grandes o suficiente.

3.3.5 Modelo de arquitetura da CNN

É possível identificar na literatura uma grande quantidade de arquiteturas de CNNs sendo utilizadas em diversas soluções de problemas que envolvem a classificação morfológica de espécies, por meio de suas imagens digitais. Na Figura 15 é possível observar graficamente estes dados.

Figura 15 – Arquitetura de CNNs

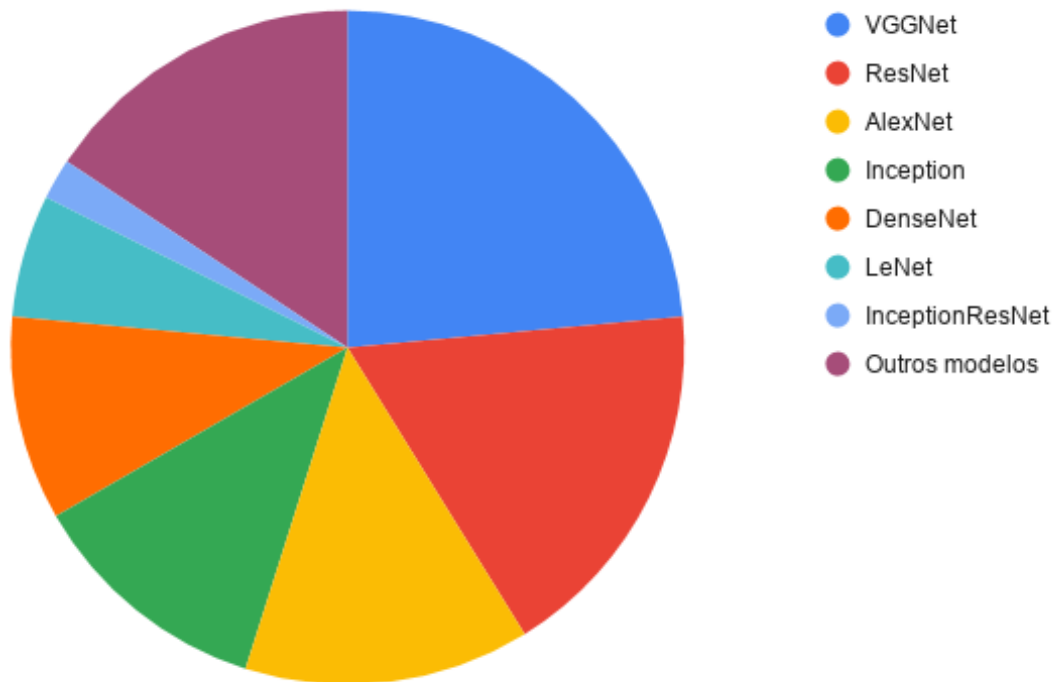


Fonte – Danilo Assunção, 2021

A VGGNet-16 (VGG16) foi usada em 14,29% (8) dos trabalhos pesquisados, dado que as arquiteturas de VGG possuem maiores profundidades com filtros convolucionais menores, podendo assim reduzir os parâmetros a serem computados, além disso, as arquiteturas VGG costumam ter bons resultados (LI *et al.*, 2020). Enquanto a AlexNet, foi aplicada em 12,50% (7) dos casos pesquisados. As arquiteturas Inception v3 e Indefinido, em que a categoria Indefinido neste contexto representa as arquiteturas implementadas pelos autores em seus artigos, representam 7,81% (5) de aplicações nos artigos selecionados nesta revisão sistemática, enquanto a ResNet-50 apresenta somente 6,25% (4) e DenseNet-121 com 4,69% (3). Já a LeNet-5, VGGNet-19 (VGG19) e InceptionResNet v2 se encontram com 3,13% (2) de aplicabilidade. Enquanto as “Outras arquiteturas” que compõem 20,31% do restante das arquiteturas utilizadas, possuem uma única utilização em seus respectivos artigos.

Para uma diferente perspectiva de visualização de usabilidade das arquiteturas de CNNs extraídas nos trabalhos em questão, foi feito um agrupamento por tipo de arquitetura para a identificação dos exemplares mais utilizados, como é mostrado na Figura 16.

Figura 16 – Tipos agrupados de arquiteturas de CNNs



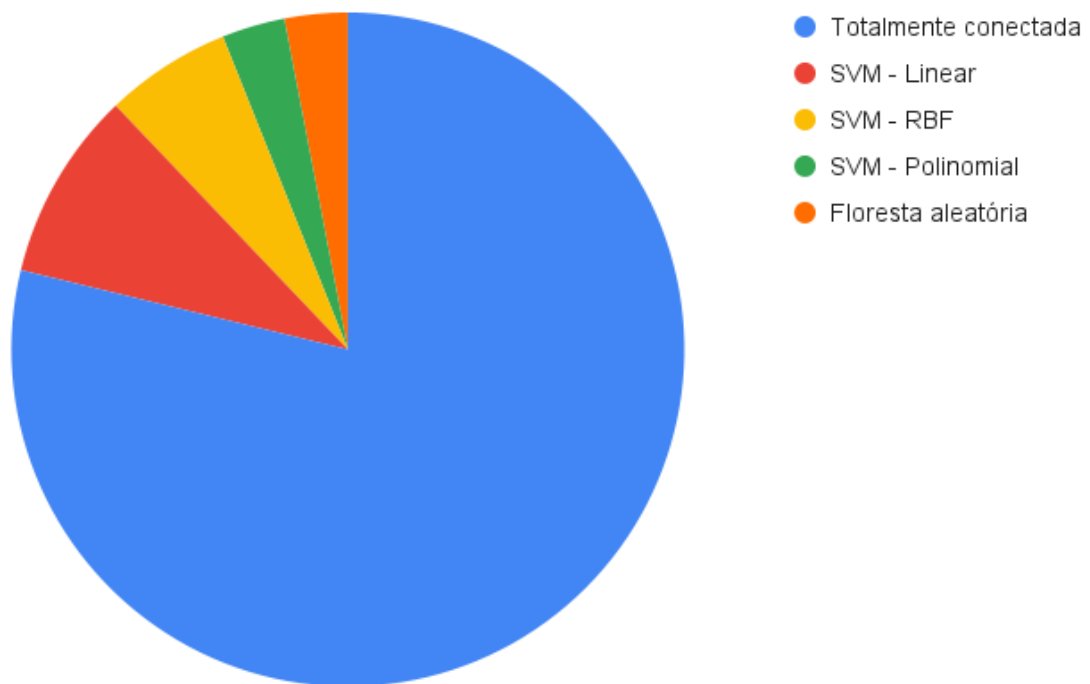
Fonte – Danilo Assunção, 2021

Nota-se que modelos de VGGNet com uso de 23,53% (12) ainda dominam em termos de arquiteturas aplicadas na resolução de problemas envolvendo a identificação de espécies. Arquiteturas do tipo ResNet ocupam o segundo lugar com 17,65% (9), isso devido ao seu comportamento de aumentar a profundidade da rede, sendo assim, as ResNets oferecem melhores recursos de aproximação de função (convergência ao resultado) à medida em que ganham mais parâmetros, contribuindo com sucesso na resolução de problemas como os de dissipação de gradiente, em que alguns modelos mais tradicionais acabam não conseguindo solucioná-lo, tendo resultados inferiores ao desejado (SUN *et al.*, 2017). Tendo a AlexNet em terceiro com 13,73% (7), sendo este tipo de arquitetura um modelo pré-treinado de rede neural convolucional que permite a aplicação de transferência de aprendizado para que atinja melhores resultados (HONG *et al.*, 2020). Modelos de Inception 11,76% (6), DenseNet 9,80% (5), LeNet 5,88% (5) e InceptionResNet 1,96% (1) são as arquiteturas que menos apareceram nos trabalhos avaliados. Por último, os “Outros modelos” que são arquiteturas de CNN desenvolvidas pelos autores dos artigos com especificidade para solucionar os seus respectivos problemas abordados.

3.3.6 Métodos de classificação

Nos trabalhos analisados é possível determinar a usabilidade de alguns tipos de algoritmos para classificação dos modelos nestes trabalhos. Dessa forma, a Figura 17 apresenta os exemplares aplicados nos artigos em questão.

Figura 17 – Método de classificação utilizado



Fonte – Danilo Assunção, 2021

A maioria das pesquisas analisadas utilizam redes totalmente conectadas, isto é, 78,79% (26) dos trabalhos e de fato tal presença deste método condiz com o padrão comumente visto na camada totalmente conectada de uma CNN. Nos demais cenários, onde a CNN é utilizada somente para extração de características, é tido a aplicação conjunta com alguns métodos (algoritmos) de classificação, sendo eles, SVM com *kernel* Linear 9,09% (3), RBF (função de base radial) 6,06% (2), Polinomial 3,03% (1) e Floresta aleatória 3,03% (1).

3.4 Conclusão

Após o término da revisão sistemática é perceptível que o uso de técnicas de aprendizado profundo, como CNNs, vem sendo empregadas em diversos cenários em que há a necessidade de otimização no processo de extração de características das imagens, para assim, solucionar problemas que envolvem reconhecimento de espécies. Um dos atributos mais importantes é a sua capacidade de aprender as características de mais alto nível de uma imagem, e fornecer prontamente uma quantidade de informações relevantes que permitam que os algoritmos de classificação possam identificar com mais eficácia o objeto visualizado. Entretanto, é importante ressaltar que o uso deste mecanismo exige uma quantidade muito grande de dados, justamente para que seu resultado venha a convergir para um modelo que tenha a capacidade de efetivar tal atividade, e para isto foi observado na literatura o uso de técnicas como transferência de aprendizado e aumento de dados que permitem a utilização de CNNs, mesmo com um conjunto pequeno de dados disponíveis e demonstrando também que o uso de pré-processamento ainda pode vir a ser extremamente útil em cenários diversos envolvendo processamento de imagens digitais.

4 Materiais e métodos

Visando a atingir o objetivo desta pesquisa, serão conduzidos experimentos com base em um conjunto de dados de treinamento, validação e teste utilizando uma arquitetura de CNN em união com técnicas de aumento de dados e transferência de aprendizado para aprimoramento dos resultados visados em cima do problema abordado. No conjunto de dados de treinamento, validação e teste, serão utilizadas as imagens de asas de abelhas obtidas em um ângulo que mostre a face da asa, tal exemplo de uma instância deste conjunto de dados pode ser observado na Figura 18. As imagens contidas nos conjuntos de dados já foram previamente rotuladas com a espécie da abelha por um especialista da área.

Figura 18 – Asa de abelha da espécie *Trigona spinipes*



Fonte – Danilo Assunção, 2021

O conjunto de dados disponibilizado é constituído por um total de 904 imagens, sendo estas imagens uma representação digital das asas de várias espécies de abelhas e de um gênero específico de vespa. A partir deste conjunto de dados é apresentado um total de 22 gêneros que agrupam as espécies de abelhas, somando um total de 45 espécies e 3 espécies de vespas do mesmo gênero. Cada espécie corresponde nesse conjunto de dados tem uma classe que será tratada no modelo em questão. Estas especificações podem ser vistas na Tabela 2.

Tabela 2 – Conjunto de dados original das asas de abelhas

Tribo	Gênero + Espécie	Quantidade de imagens
Euglossini	<i>Eufriesea violacea</i>	20
	<i>Euglossa mixta</i>	20
	<i>Euglossa annectans</i>	19
	<i>Euglossa truncata</i>	15
	<i>Eulaema nigrita</i>	10
	<i>Exaerete smaragdina</i>	8
Apini	<i>Apis cerana</i>	20
	<i>Apis dorsata</i>	20
	<i>Apis florea</i>	20
	<i>Apis mellifera</i>	19
Bombini	<i>Bombus (Fervidobombus) brasiliensis</i>	20
	<i>Bombus (Fervidobombus) pauloensis</i>	11
Meliponini	<i>Austroplebeia australis</i>	18
	<i>Austroplebeia cincta</i>	20
	<i>Austroplebeia essinatoni</i>	20
	<i>Austroplebeia striped</i>	20
	<i>Austroplebeia symei</i>	20
	<i>Axestotrigona ferrugínea</i>	20
	<i>Dactylurina staudingeri</i>	20
	<i>Geotrigona sp</i>	16
	<i>Lestrimelitta limao</i>	19
	<i>Melipona (Eomelipona) bicolor</i>	14
	<i>Melipona mandacaia</i>	20
	<i>Melipona quadrifasciata</i>	20
	<i>Melipona subnitida</i>	19
	<i>Melipona (Michmelia) flavolineata</i>	20
	<i>Melipona (Michmelia) scutellaris</i>	20
	<i>Melipona seminigra seminigra</i>	20
	<i>Meliponula bocandei</i>	20
<i>Mourella caerulea</i>	20	

	<i>Nannotrigona testaceicornis</i>	20
	<i>Paratrigona subnuda</i>	20
	<i>Partamona helleri</i>	20
	<i>Plebeia droryana</i>	20
	<i>Plebeia flavocincta</i>	19
	<i>Plebeia nigriceps</i>	20
	<i>Plebeia pugnax</i>	20
	<i>Plebeia remota</i>	20
	<i>Plebeia sp</i>	20
	<i>Scaptotrigona bipunctata</i>	20
	<i>Scaptotrigona depilis</i>	20
	<i>Scaptotrigona tubiba</i>	19
	<i>Scaura latitarsis</i>	20
	<i>Schwarziana quadripunctata</i>	19
	<i>Trigona spinipes</i>	20
Trypoxylini	<i>Trypoxylon aurifrons</i>	20
	<i>Trypoxylon lactitarse</i>	20
	<i>Trypoxylon rogenhoferi</i>	20
Fonte - Danilo Assunção, 2022		

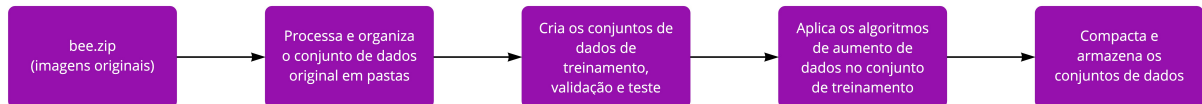
Conforme apresentado, para levantar o estado da arte para o uso de uma arquitetura de CNN em união com uma técnica de transferência de aprendizado e técnicas de aumento de dados para realizar a identificação automática de espécie de abelhas, foi elaborada uma revisão sistemática utilizando os conceitos aplicados em [Liu et al. \(2020b\)](#).

Após o levantamento do estado da arte, foi construída uma ferramenta (*pipeline*) constituída de dois módulos essenciais, o primeiro módulo sendo o responsável pela geração dos conjuntos de dados de treinamento, validação e teste em cima do conjunto de dados original formado pelas 904 imagens, em que o conjunto de treinamento terá seu volume aumentado através do uso de técnicas de aumento de dados, e o segundo módulo que aplicará a configuração, seleção da arquitetura de CNN e execução do modelo para ser trabalhado em cima dos conjuntos de dados organizados pelo primeiro módulo.

4.1 Módulo do conjunto de dados

Este primeiro módulo consiste em extrair, organizar, dividir, compactar e armazenar o conjunto de dados criado. Os passos executados por este módulo podem ser vistos na Figura 19.

Figura 19 – Representação do módulo do conjunto de dados



Fonte – Danilo Assunção, 2022

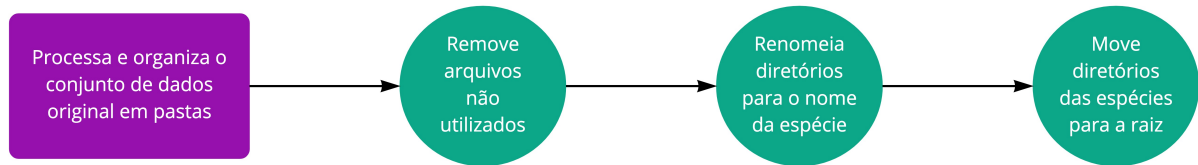
Conforme exibido na Figura 19, este módulo foi dividido em cinco etapas de execução, onde cada quadrado roxo representa uma etapa. Estas etapas são definidas como:

1. Carregamento dos dados originais;
2. Processamento e organização do conjunto de dados;
3. Divisão dos conjuntos de dados;
4. Aumento do conjunto de dados de treinamento;
5. Compactação e armazenamento dos conjuntos de dados criados.

No carregamento dos dados originais, é feito o carregamento do conjunto de dados originais contendo as 904 imagens que será transformada em uma representação computacional a qual a ferramenta que irá processar o conteúdo possa computar tais imagens.

Na etapa de processamento e organização do conjunto de dados, as imagens previamente carregadas foram organizadas automaticamente em uma estrutura de diretórios, em que cada diretório representa uma determinada espécie de abelha contendo suas respectivas imagens, conforme apresentado na Figura 20.

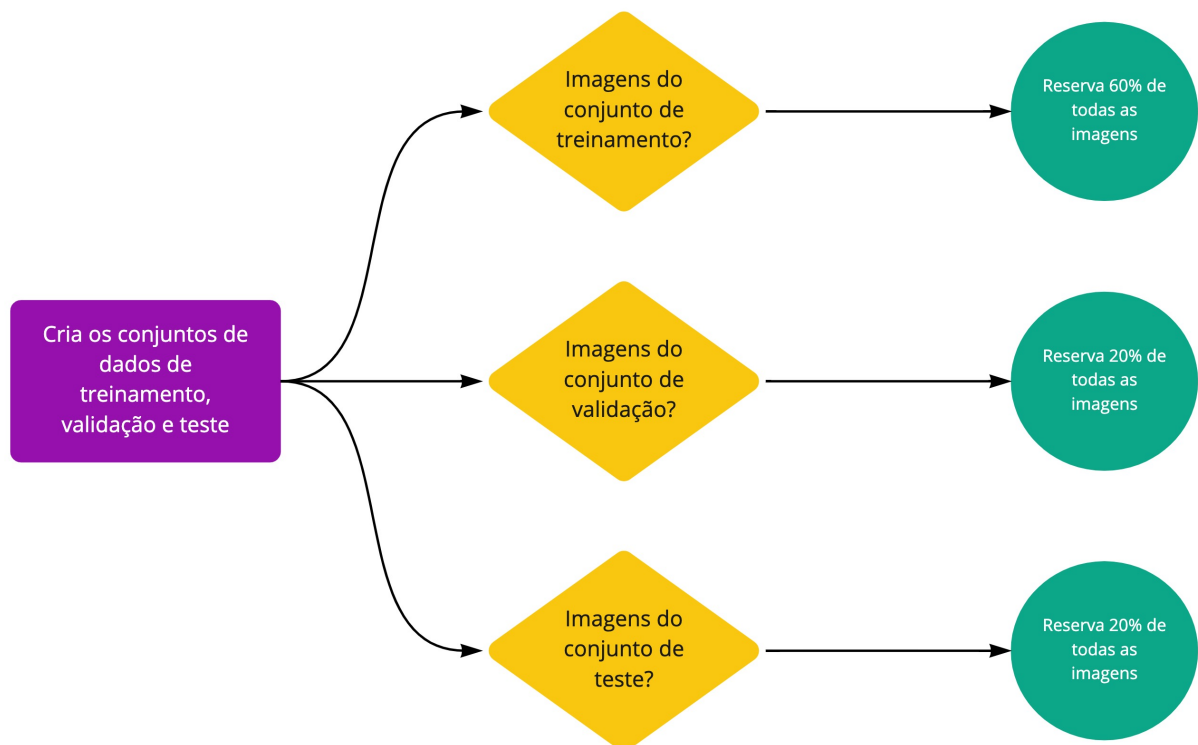
Figura 20 – Processamento e organização das imagens



Fonte – Danilo Assunção, 2022

Na etapa de divisão dos conjuntos de dados, é feito um agrupamento de um número determinado pelo percentual com base no total de imagens disponíveis para diretórios que representam o conjunto de treinamento, validação e teste. Tal divisão é feita selecionando 60% das imagens disponíveis para o conjunto de dados de treinamento, 20% das imagens para o conjunto de dados de validação e por fim os últimos 20% das imagens restantes são direcionadas para o conjunto de dados de teste, esta representação pode ser vista na Figura 21.

Figura 21 – Divisão dos conjuntos de dados

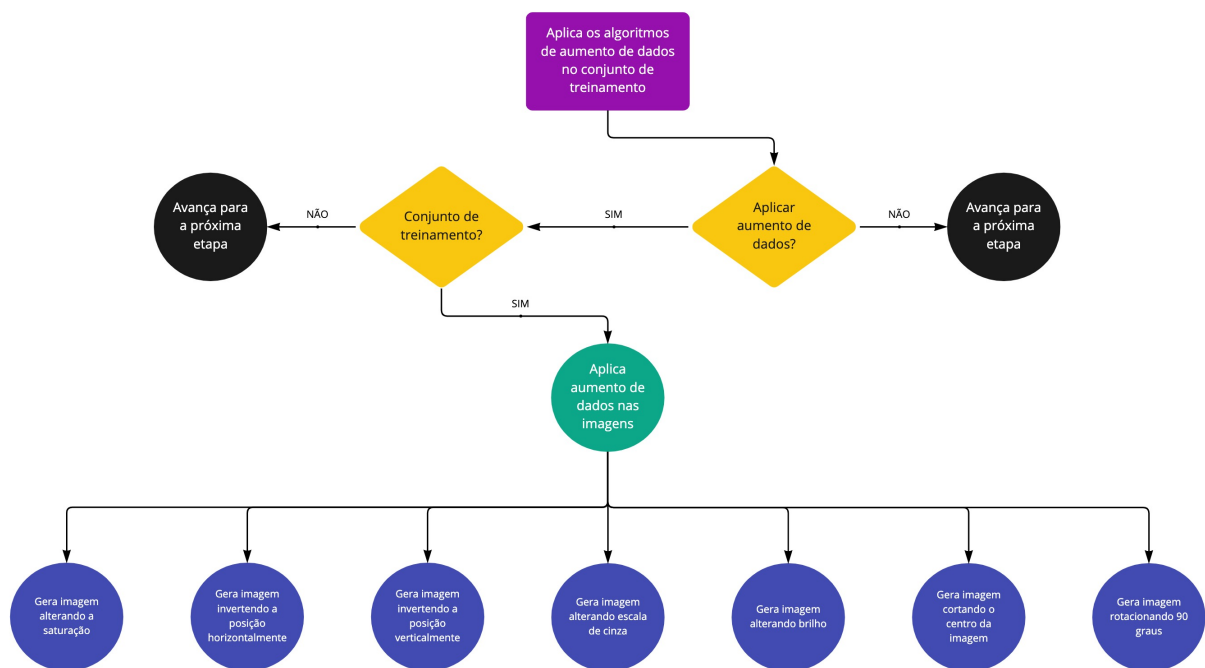


Fonte – Danilo Assunção, 2022

Na etapa de aumento do conjunto de dados de treinamento, é feito a aplicação de algoritmos de aumento de dados no conjunto de dados de treinamento e este procedimento é feito aplicando algoritmos que cria uma nova imagem alterando a propriedade base

da imagem original utilizando métodos como, alteração do nível de saturação, inversão horizontal, inversão vertical, alteração na escala de cinza, alteração do brilho, corte central e rotação de 90 graus e estes métodos permitem que todo o conjunto de dados de treinamento seja multiplicado em 7 vezes o seu tamanho original seja replicada em mais outras 7 imagens. Este método permitiu que o conjunto de dados de treinamento original aumentasse de 544, sendo este número de imagens o resultado do arredondamento que representa os 60% do conjunto de dados original, para 3.808 imagens e tais métodos aplicados para este aumento representados pelos círculos azuis na Figura 22 são categorizados como métodos de transformação geométrica e foram selecionados para uso neste trabalho devido a sua facilidade de implementação (SHORTEN; KHOSHGOFTAAR, 2019). O procedimento completo desta etapa pode ser visto na Figura 22.

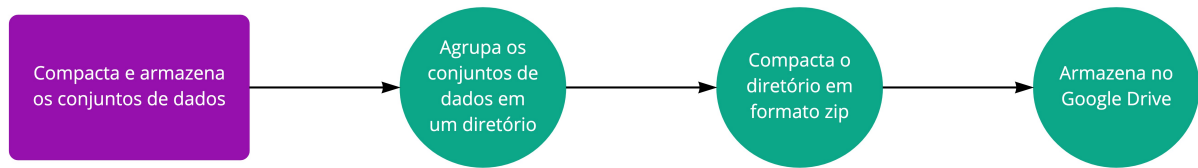
Figura 22 – Aumento do conjunto de dados de treinamento



Fonte – Danilo Assunção, 2022

Na etapa de compactação e armazenamento do conjunto de dados, é realizado o agrupamento de todos os conjuntos de dados criados em um diretório onde tal diretório é compactado e armazenado na nuvem (*Google Drive*). Esses passos podem ser vistos na Figura 23.

Figura 23 – Compactação e armazenamento do conjunto de dados

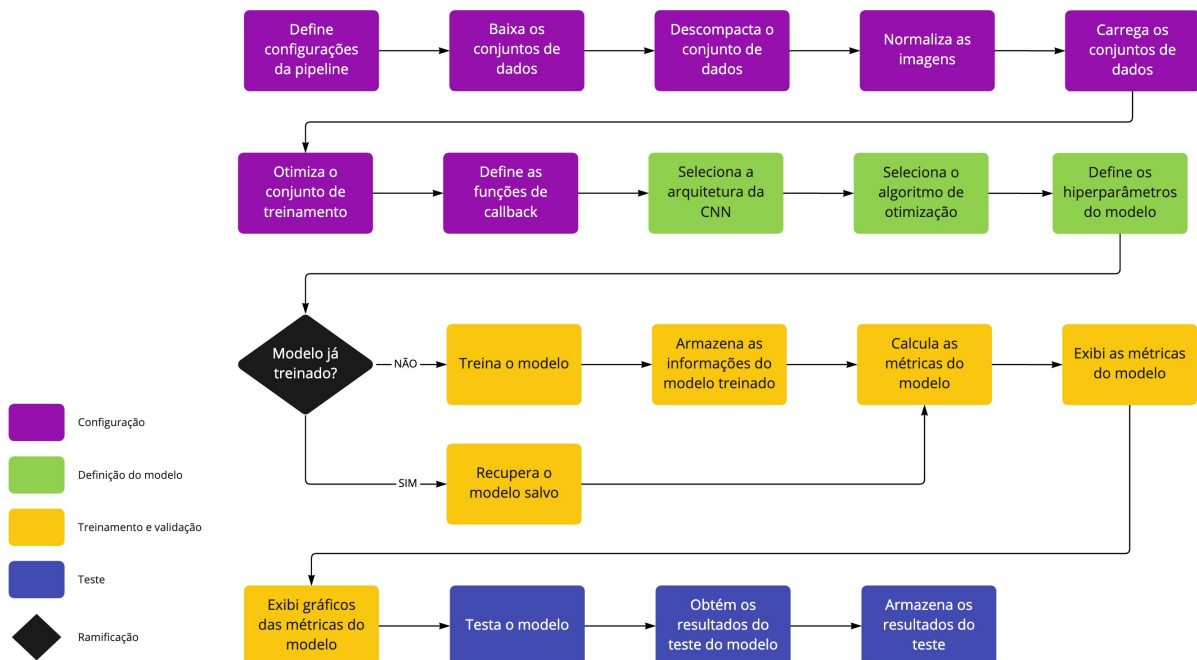


Fonte – Danilo Assunção, 2022

4.2 Módulo do modelo de classificação

O módulo do modelo de classificação possui a responsabilidade de consumir o conjunto de dados pré-processado pelo módulo do conjunto de dados, configurar, selecionar uma arquitetura de CNN e executá-la com o conjunto de dados carregado, testar e exibir os resultados de todos os passos executados. Como é possível observar através da Figura 24, este módulo é dividido em quatro grupos de execução, sendo estes grupos, Configuração; Definição do modelo; Treinamento e validação e por fim o Teste.

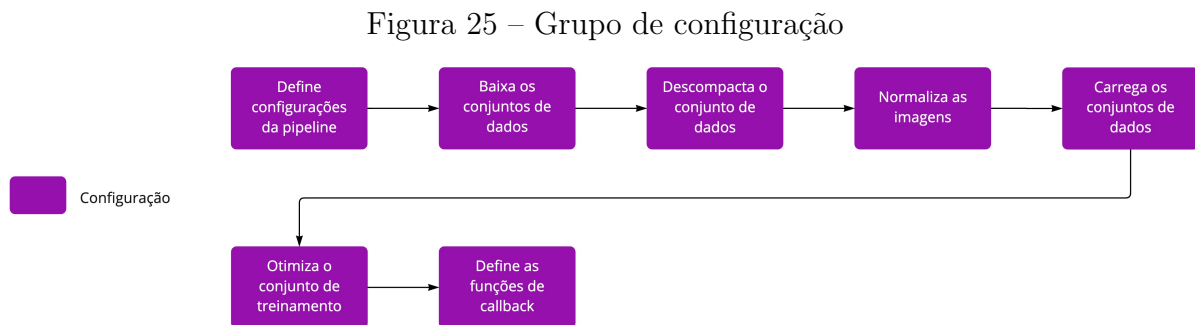
Figura 24 – Representação do módulo do modelo de classificação



Fonte – Danilo Assunção, 2022

4.2.1 Grupo de configuração

O grupo de configuração tem como objetivo determinar todos os parâmetros necessários deste módulo e, também, extrair e carregar o conjunto de dados pré-processado, otimizar e configurar funções de *callback* (funções personalizadas chamadas durante o processo de execução), como pode ser visto na Figura 25.



Fonte – Danilo Assunção, 2022

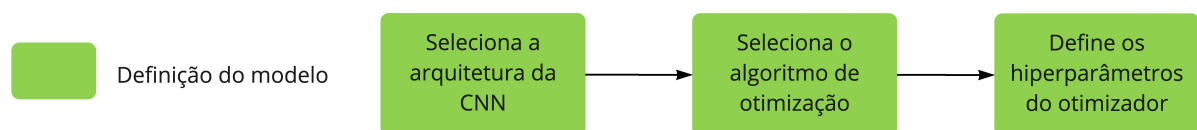
As configurações do *pipeline* são constantes definidas contendo uma série de características presentes neste módulo, como a dimensão das imagens que serão trabalhadas, a localização de diretórios locais e na nuvem, e a parametrização para as condicionais executadas em pontos específicos, tendo como exemplo a exibição de *logs* da ferramenta e alguns parâmetros gerais utilizados pelo modelo, tal como, o número de *batches* por época. Então, é feito a transferência do conjunto de dados pré-processados através do *Google Drive*, para assim o conjunto de dados ser descompactado e gerar três diretórios, contendo, cada um, o conjunto de dados de treinamento, validação e teste. Após a obtenção dos conjuntos de dados distintos, é realizado um processo de normalização nos *pixels* das imagens de treinamento para valores compreendidos entre 0 e 1. Em seguida, os dados são carregados em estruturas que redimensionam e organizam as imagens em memória RAM, e, estas estruturas, são criadas e divididas em três partições, a de treinamento, validação e teste, em que, esses conjuntos de dados acabam por ter nestas estruturas a aplicação de um algoritmo que aleatoriamente embaralha as imagens contidas em cada um dos conjunto de dados. Após a criação destas estruturas, os dados presentes para treinamento são otimizados utilizando um mecanismo de *cache*, para que o processo de leitura das imagens seja realizado de maneira mais otimizada a cada *batch* de imagens que serão entregues ao ciclo de treinamento do modelo, e, por fim, algumas funções (*callback functions*) são

definidas para serem chamadas durante a execução do treinamento, sendo elas responsáveis por recuperar os resultados obtidos em cada época no processo de treinamento do modelo.

4.2.2 Grupo de definição do modelo

No grupo de definição do modelo, temos os passos responsáveis por selecionar a arquitetura de CNN que é utilizada na execução do modelo, sendo esta metodologia empregada visando a determinar dinamicamente as arquiteturas que serão utilizadas nos testes do modelo determinado. As arquiteturas de CNN foram escolhidas com base na revisão sistemática realizada neste trabalho, sendo elas, InceptionResnet v2, VGGNet 16, VGGNet 19, Resnet-152, Inception v3. Cada uma destas arquiteturas pode ou não utilizar uma técnica de transferência de aprendizado de ajuste fino, e para o uso desta técnica, será necessário o uso de pesos já pré-treinados. O método de ajuste fino foi escolhido como o método de transferência de aprendizado deste trabalho dado os aspectos positivos em seu uso especificados em [Li e Hoiem \(2017\)](#). Em seguida, é feita a seleção dos algoritmos de otimização da CNN, tais algoritmos foram, Gradiente Descendente Estocástico (SGD), Adadelta, Adagrad, Adamax. Posteriormente é feita a definição dos hiperparâmetros do otimizador selecionado, para assim os parâmetros serem igualmente definidos para todas as arquiteturas propostas, tendo como exemplo a taxa de aprendizado. A representação desta etapa pode ser visualizada através da [Figura 26](#).

Figura 26 – Grupo de definição do modelo

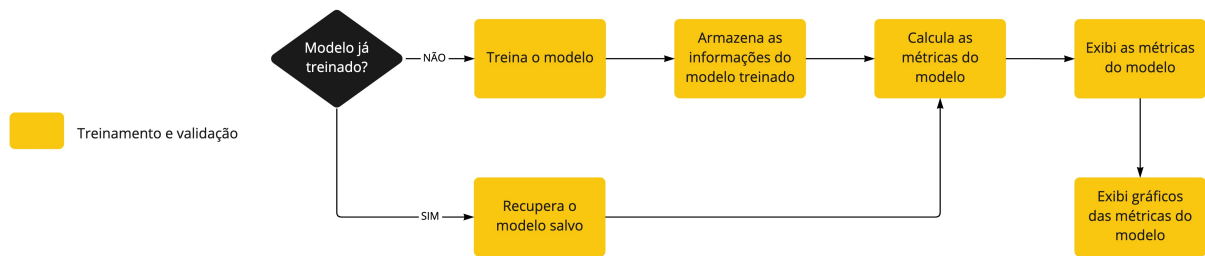


Fonte – Danilo Assunção, 2022

4.2.3 Grupo de treinamento e validação

Este grupo possui as etapas responsáveis por realizar o fluxo de treinamento e validação do modelo utilizando os conjuntos de treinamento e validação para, respectivamente, obter e armazenar os resultados deste processamento e exibir algumas representações gráficas dos resultados retornados. O fluxo de todo o processo pode ser visto na [Figura 27](#).

Figura 27 – Grupo de treinamento e validação



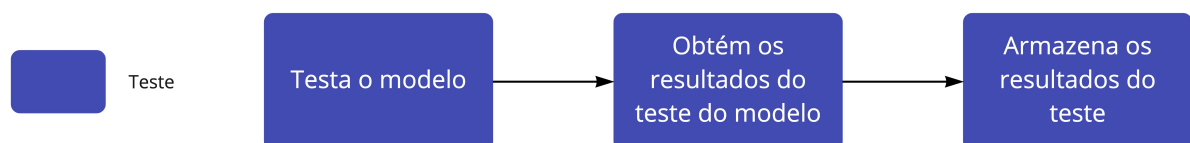
Fonte – Danilo Assunção, 2022

O fluxo de treinamento e validação realiza uma checagem, na qual consiste em validar se as configurações definidas para o algoritmo do modelo selecionado já possui dados salvos de treinamento, caso houver, é feita a recuperação destes dados do modelo já treinado na nuvem, para assim seguir para o passo de cálculo e exibição das métricas do treinamento, caso contrário é realizado o treinamento do modelo e executado os passos seguintes. As métricas do processo de treinamento são exibidas durante cada época e, no final de todo o treinamento, é feita a coleta das medidas resultantes deste processo, sendo essas medidas a Precisão, Revocação, Acurácia e Taxa de erro. Com a visualização gráfica desses resultados, são geradas duas representações, sendo elas, o gráfico de acurácia por épocas e o gráfico de perda por épocas (dados da função de perda).

4.2.4 Grupo de teste

Neste grupo o modelo é testado utilizando o conjunto de dados de teste a partir do modelo treinado. Dado esta análise, logo em seguida é realizado o cálculo das medidas como Precisão, Revocação, Acurácia, Medida F e a Matriz Confusão. Por fim, estes resultados são armazenados em estruturas organizadas no *Google Drive*. A representação do fluxo das etapas do grupo de teste pode ser visto através da Figura 28.

Figura 28 – Grupo de teste

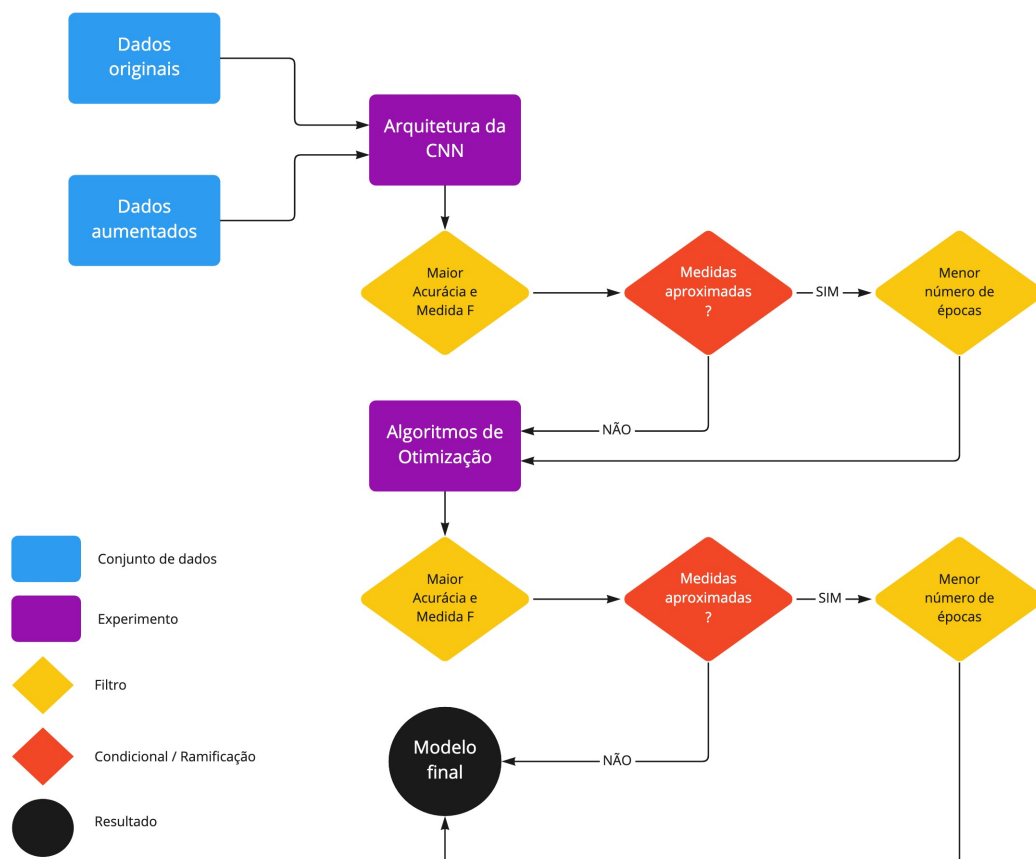


Fonte – Danilo Assunção, 2022

4.3 Avaliação

A avaliação consiste em comparar os resultados produzidos automaticamente pela CNN e, assim, observar a aplicação de métricas de aprendizado de máquina supervisionado (Revocação, Precisão, Acurácia, Medida F), possibilitando mensurar os resultados obtidos no processo de discriminação das classes, com o objetivo de obter o estado da arte para o modelo adequado ao problema proposto. A estratégia para esta avaliação será feita a partir de determinados passos, representados pela Figura 29.

Figura 29 – Estratégia de experimentação



Fonte – Danilo Assunção, 2022

A estratégia de experimentação estabelecida é ilustrada na Figura 29, em que o processo realiza a seleção da arquitetura de CNN e do algoritmo de otimização para a CNN que demonstrar o melhor resultado, baseando-se no modelo abordado e seguindo as etapas pré-estabelecidas.

Serão utilizados dois conjuntos de dados para essa experimentação, sendo eles, o conjunto com as imagens originais e o conjunto com as imagens aumentadas pela técnica

de aumento de dados. Para cada conjunto, foi feito a atribuição de etapas contendo uma experimentação que consiste na execução de um mecanismo que produz métricas de comparação, e um filtro que determina o item que será utilizado na próxima etapa do fluxo de experimentação. A primeira etapa consiste em determinar qual arquitetura de CNN será selecionada com base nas arquiteturas propostas, e, para todos estes exemplares propostos, foram considerados os hiperparâmetros relacionados em [Buschbacher *et al.* \(2020\)](#). Com ênfase nas técnicas e experiências de tal trabalho, foi feito o uso de SGD como algoritmo de otimização com um valor de momento (técnica incluída no SGD que acumula o gradiente dos passos anteriores para determinar a direção a seguir) de 0,9 e com uma taxa de aprendizado de 0,001 para até 30 épocas, 0,0005 de 31 a 40 épocas e 0,0001 de 41 a 50 épocas, que será o número total de épocas aplicada nos experimentos, e, para as arquiteturas de CNN propostas, será feito um caso de teste com ou sem o uso de uma técnica de transferência de aprendizado. Após a realização das experimentações citadas anteriormente, os valores obtidos passaram por um filtro que determinará qual a arquitetura de CNN empregada no modelo que alcançou a maior Acurácia e Medida F. Durante o experimento de seleção da CNN, é considerada uma regra de captura destes resultados, realizando o armazenamento dos dados do modelo para uma determinada época, sendo isso apontado com base em uma condição de parada, que é acionada caso após as cinco épocas de treinamento seguintes do modelo não houver melhorias em seu desempenho. Após a execução deste primeiro experimento é feita uma validação e, assim, avaliar a existência de testes com resultados aproximados, dado este caso, será feito o uso de um segundo filtro, determinando a melhor arquitetura de CNN do modelo com base no número de épocas usado em treinamento para os modelos treinados nesta etapa.

Para a última etapa do experimento, foi considerada a escolha do algoritmo de otimização mais adequado para a arquitetura de CNN selecionada no passo anterior, em que serão realizados experimentos utilizando a arquitetura selecionada através do primeiro experimento, porém, realizado os testes a partir da utilização dos algoritmos de otimização propostos. A seleção do algoritmo de otimização foi definido ao filtrar o cenário que tiver como resultado a maior Acurácia e Medida F entre eles. Em adicional, caso existam cenários com resultados aproximados, foi considerado a aplicação de mais um filtro, considerando o número de épocas exigidas para se obter o melhor modelo deste experimento.

5 Resultados e discussões

Este capítulo detalha os resultados obtidos durante o processo de treinamento, validação e teste do modelo de aprendizado de máquina utilizando algoritmos de CNN e otimização que foram selecionados para a resolução do problema em questão, que envolve a identificação automática de espécies de abelhas através de imagens de suas asas.

Seguindo as diretrizes estabelecidas pela metodologia especificada, é feito o detalhamento dos resultados e a exemplificação de cada cenário de maneira descritiva e ilustrativa a partir dos dados coletados nesta etapa. Dito isto, as seções a seguir seguirão o detalhamento dos passos descritos pela Figura 29, considerando as etapas de experimentação, em que é sucedida a primeira etapa para seleção da arquitetura de CNN, e, por fim, a seleção do algoritmo de otimização para a CNN.

5.1 Seleção da arquitetura da CNN

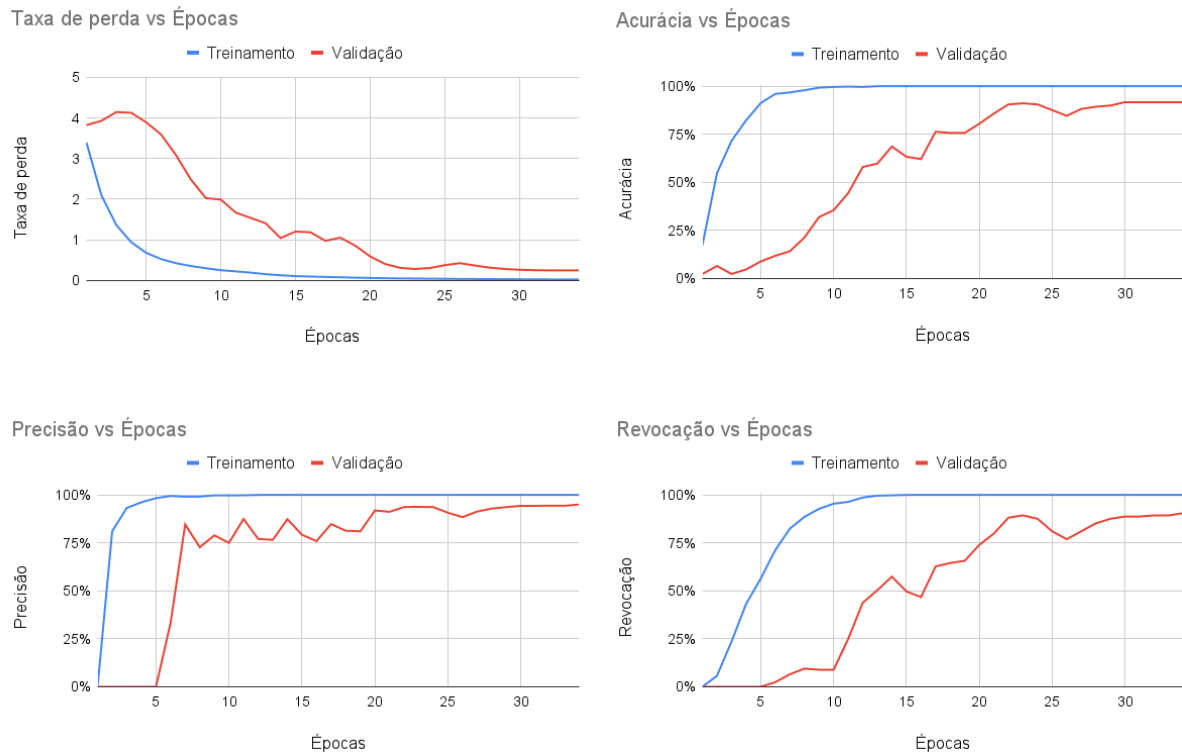
Esta seção tem como objetivo, detalhar os experimentos realizados no processo de seleção da arquitetura da CNN que obteve o melhor resultado com relação ao domínio proposto. As arquiteturas de CNN abordadas neste trabalho foram selecionadas com base na revisão sistemática feita também neste trabalho, sendo elas, InceptionResNet v2, Inception v3, ResNet-152, VGG16 e VGG19. Tais arquiteturas também foram empregadas levando em consideração o uso de técnicas de aumento de dados e transferência de aprendizado para fins comparativos.

5.1.1 InceptionResNet v2

Para o modelo empregando a arquitetura de CNN InceptionResNet v2, o primeiro experimento realizado foi utilizando o conjunto de dados original, que contém as imagens disponibilizadas sem o uso de técnicas de aumento de dados. Este modelo foi treinado por até 34 épocas, posteriormente, não houve melhorias em seus resultados, considerando o critério de parada que são 5 épocas no total. Para os experimentos utilizando apenas InceptionResNet v2, não foi aplicada nenhuma técnica de transferência de aprendizado.

A seguir, é possível visualizar os resultados obtidos na fase de treinamento para este experimento, através da Figura 30.

Figura 30 – Resultados do treinamento com a InceptionResNet v2 + Dados originais



Fonte – Danilo Assunção, 2022

As representações da Figura 30 representam as linhas de treinamento, que correspondem aos dados de treinamento obtidos em cada época do modelo e respectivamente a linha que representa os dados obtidos em cada época na validação do modelo e todos os gráficos representam, cada um deles, uma medida obtida para determinar o desempenho geral do modelo, sendo eles, respectivamente, a taxa de perda, acurácia, precisão e revocação resultantes deste processo. A taxa de perda é um número que mensura o fator de previsibilidade do modelo, ou seja, caso a previsão do modelo for boa, a perda tende à zero, caso contrário, a perda será maior.

Esta arquitetura é a união de duas arquiteturas consistentes que são a Inception e a ResNet, e, esta junção, visa a obter algumas características relevantes de ambas as arquiteturas, isto é, a família de arquiteturas Inception são exemplares que demonstram alcançar um desempenho muito bom a um custo computacional relativamente baixo (SZEGEDY *et al.*, 2017), enquanto a arquiteturas da família ResNet é uma abordagem

que faz uso de conexões de atalho, e este aspecto auxilia na resolução do problema de Dissipação do Gradiente (FENG, 2017), apontando que, no decorrer do treinamento de uma rede, os gradientes propagados passarão por multiplicações com valores menores que 1, realizados em cada camada da rede e possuindo valores mínimos ao chegar nas camadas iniciais da rede. Tendo como efeito, o encontro também de valores mínimos pelo ajuste dos pesos, sendo ele calculado a cada iteração do treinamento da rede. Desta forma, devido ao requisito de uma elevada quantidade de iterações para se obter um ajuste minimamente considerável, acaba por acarretar em um alto custo para o treinamento das primeiras camadas da rede (HOCHREITER, 1998). Diante disso, a Dissipação do Gradiente soluciona problemas relacionados quando se possui redes muito profundas, dado que, a adição de mais camadas na rede permite determinar representações mais complexas e, para problemas mais complexos, possuir uma CNN profunda pode ser necessário. Para este modelo, foram obtidos os resultados e coletadas as suas medidas, sendo elas apresentadas no Quadro 1.

Quadro 1 – Resultados do teste com a InceptionResNet v2 + Dados originais

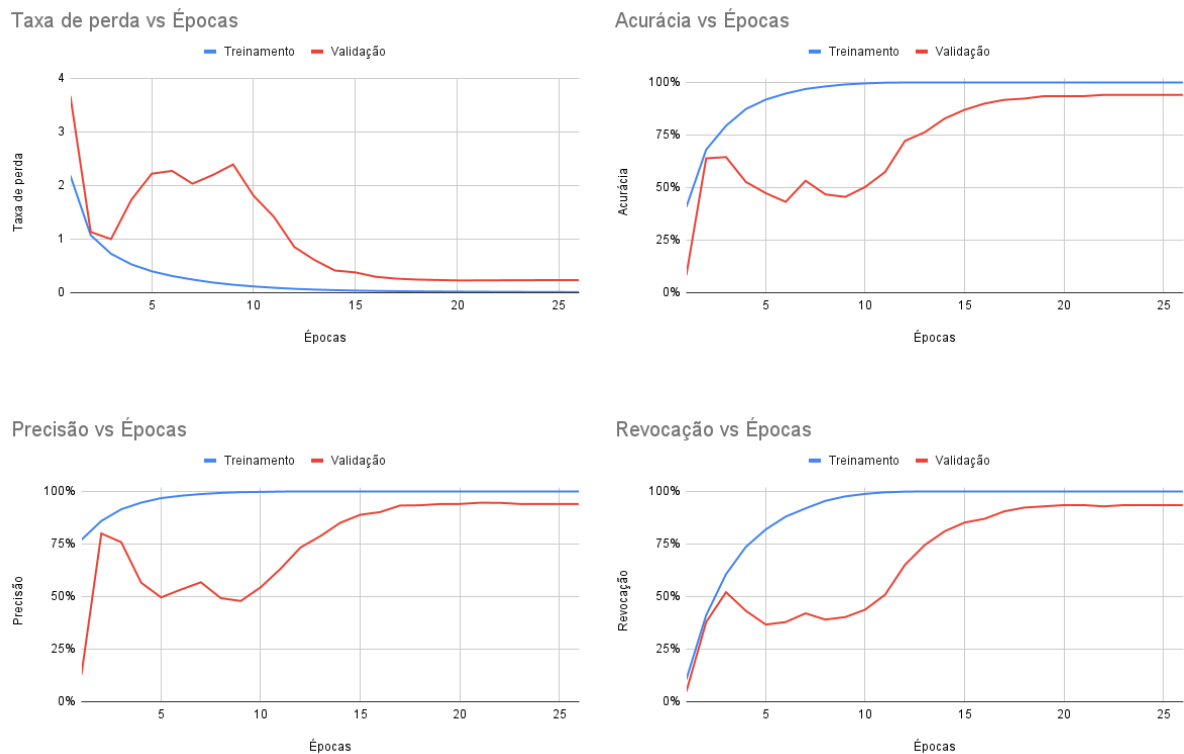
Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
34	0,24	90,58%	93,30%	87,43%	90,27%	173	18

Fonte – Danilo Assunção, 2022

No teste deste experimento foi alcançado um valor de 90,27% para a Medida F, que é a medida sendo utilizada para comparativo neste trabalho e, considerando que este modelo não possui nenhuma técnica de aumento de dados e transferência de aprendizado, ele conseguiu atingir um valor de 90,58% de Acurácia, acertando um total de 173 imagens e errando 18 imagens.

Ainda visando melhorias no modelo atual, foi efetuada uma experimentação utilizando a mesma arquitetura de CNN, mas aplicando em adicional a técnica de aumento de dados ao conjunto de dados original, obtendo-se assim um conjunto de dados aumentado em até 7 vezes o seu tamanho original. Neste experimento, considerando o conjunto de dados aumentado, o modelo teve de treinar durante um total de 26 épocas, até não demonstrar mais melhorias e aplicar o critério de parada do processo de treinamento. Os dados adquiridos no treinamento deste modelo podem ser vistos na Figura 31.

Figura 31 – Resultados do treinamento da InceptionResNet v2 + Dados aumentados



Fonte – Danilo Assunção, 2022

Para a fase de teste deste modelo utilizando os dados aumentados, temos a visualização dos resultados obtidos no Quadro 2.

Quadro 2 – Resultados do teste com a InceptionResNet v2 + Dados aumentados

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
26	0,13	94,24%	95,21%	93,72%	94,46%	180	11

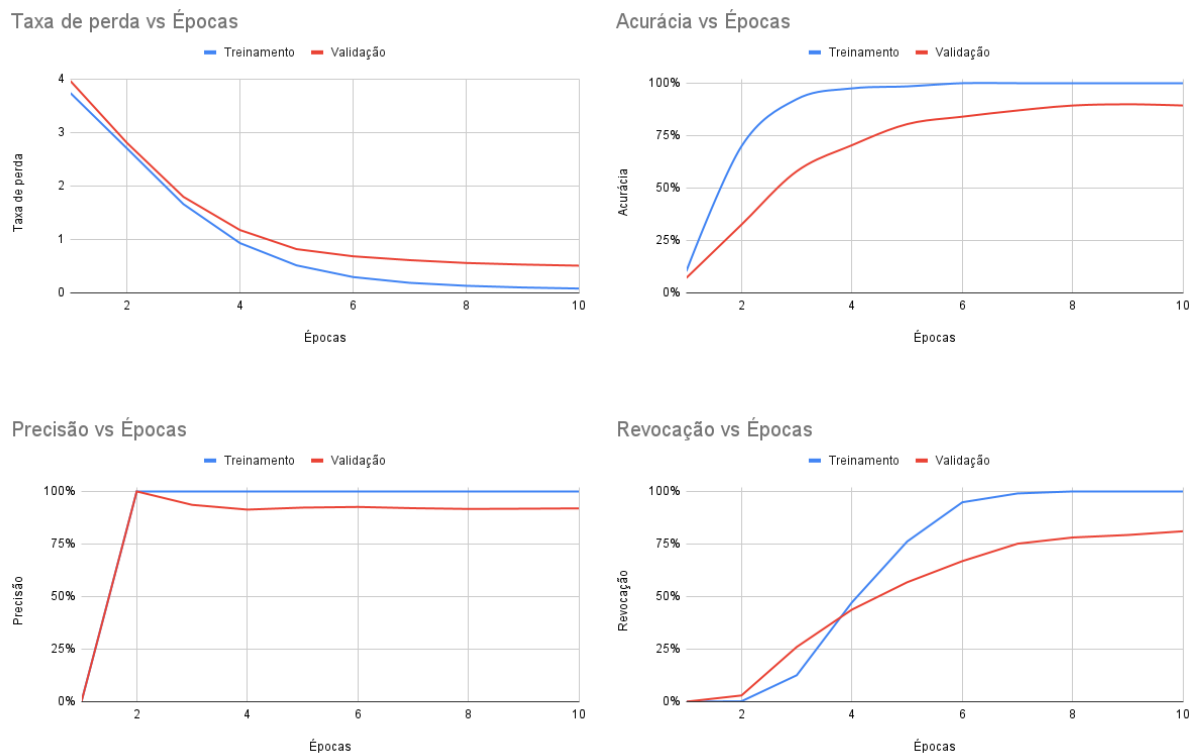
Fonte – Danilo Assunção, 2022

Comparando a Medida F e a Acurácia resultante dos dois últimos testes, o modelo que utilizou o conjunto de dados aumentado teve um acréscimo de 4,19% em sua Medida F e 3,66% em sua Acurácia, somente ampliando a quantidade de imagens do conjunto de dados. É observável também que, além do desempenho ter sido melhorado com o aumento de dados, a quantidade de épocas necessárias para se ter este resultado foi menor com relação ao modelo que não usa o aumento de dados, tendo utilizado 8 épocas a menos para se atingir resultados melhores neste modelo. Sendo assim, o uso da InceptionResNet v2 com aumento de dados, mostrou ser melhor neste experimento.

Considerando os modelos anteriores, foram também executados testes utilizando o conjunto de dados original e o conjunto de dados aumentado, junto à técnica de

transferência de aprendizado. O primeiro teste foi feito empregando apenas o conjunto de dados original e seus dados obtidos no processo de treinamento, podendo ser observado na Figura 32.

Figura 32 – Resultados do treinamento com a InceptionResNet v2 + Dados originais + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Para o teste deste modelo foi coletado os valores que podem ser vistos através do Quadro 3.

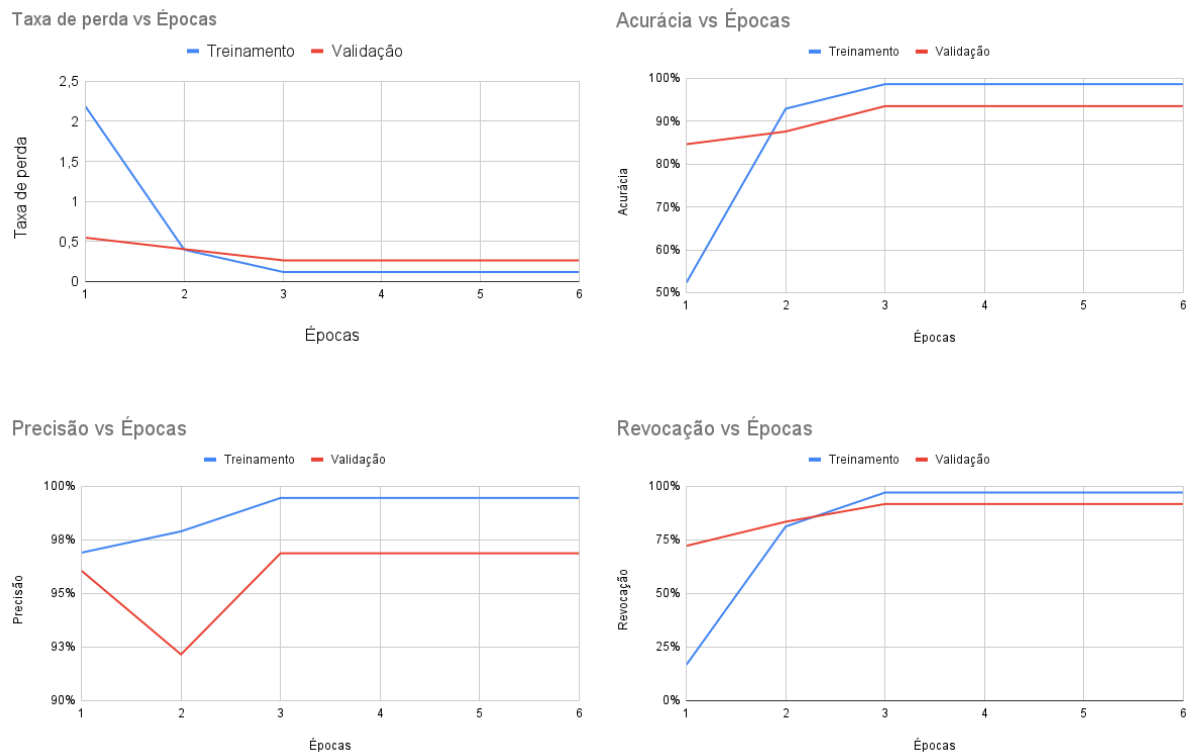
Quadro 3 – Resultados do teste com a InceptionResNet v2 + Dados originais + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
10	0,47	90,58%	93,90%	80,63%	86,76%	173	18

Fonte – Danilo Assunção, 2022

Por fim, o último experimento utilizando esta arquitetura foi feito aplicando ambas as técnicas de aumento de dados e de transferência de aprendizado. Os resultados obtidos durante o treinamento deste modelo, usando ambas as técnicas, podem ser analisados através da Figura 33.

Figura 33 – Resultados do treinamento com a InceptionResNet v2 + Dados aumentados + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

O treinamento deste modelo, logo nas épocas iniciais, já demonstrava relativamente bons resultados devido ao uso da técnica de transferência de aprendizado. Em relação ao modelo que usa transferência de aprendizado com os dados originais, este exemplo precisou de apenas 6 épocas, tendo obtido seu melhor resultado em 4 épocas a menos em comparação com o modelo anterior. Os dados do teste deste experimento podem ser visualizados no Quadro 4.

Quadro 4 – Resultados do teste com a InceptionResNet v2 + Dados aumentados + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
5	0,31	91,62%	95,58%	90,58%	93,01%	175	16

Fonte – Danilo Assunção, 2022

Os resultados em termos de desempenho deste último modelo, demonstraram melhores medidas em comparação com a Medida F e Acurácia do modelo que usa a técnica de transferência de aprendizado com o conjunto de dados original, tendo obtido uma

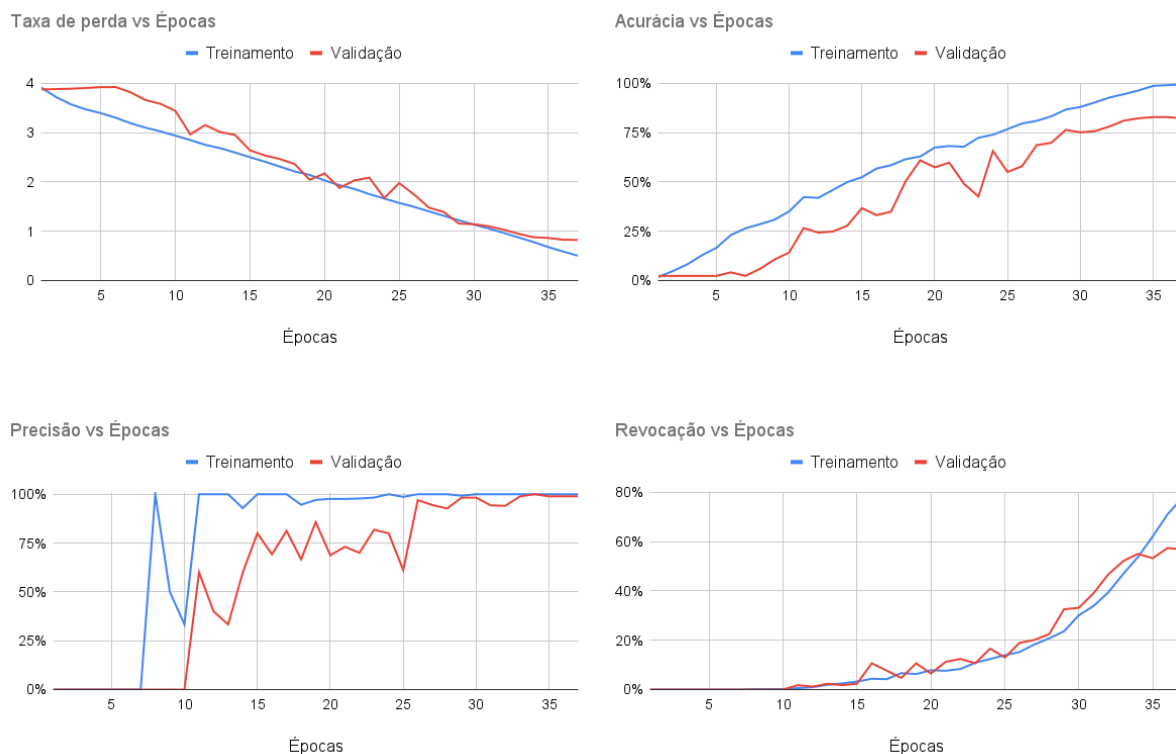
melhoria de 6,25% em sua Medida F e 1,04% na Acurácia utilizando 5 épocas a menos do que o modelo anterior.

Comparando os dois experimentos com maiores resultados, sendo eles, este último experimento e o experimento que utiliza apenas o conjunto de dados aumentado, em que, o experimento que aplica somente a técnica de aumento de dados foi superior a este último modelo experimentado, atingindo uma diferença de 1,45% em sua Medida F e 2,62% em sua Acurácia, entretanto utilizando 21 épocas a mais do que o modelo comparado.

5.1.2 Inception v3

Inception v3 é uma arquitetura de rede neural convolucional da família Inception que possui certas melhorias com relação a suas versões antecessoras, e foi construída com base em [Szegedy *et al.* \(2016\)](#). Para o modelo que usa a arquitetura Inception v3, na realização do primeiro experimento foi utilizando o conjunto de dados original. Este modelo foi treinado por 37 épocas até não haver mais melhorias em seus resultados. A seguir, é possível visualizar os resultados obtidos na fase de treinamento para este experimento, através da [Figura 34](#).

Figura 34 – Resultados do treinamento com a Inception v3 + Dados originais



Fonte – Danilo Assunção, 2022

Para este modelo, foram obtidos os resultados e coletadas as suas medidas, sendo elas apresentadas no Quadro 5.

Quadro 5 – Resultados do teste com a Inception v3 + Dados originais

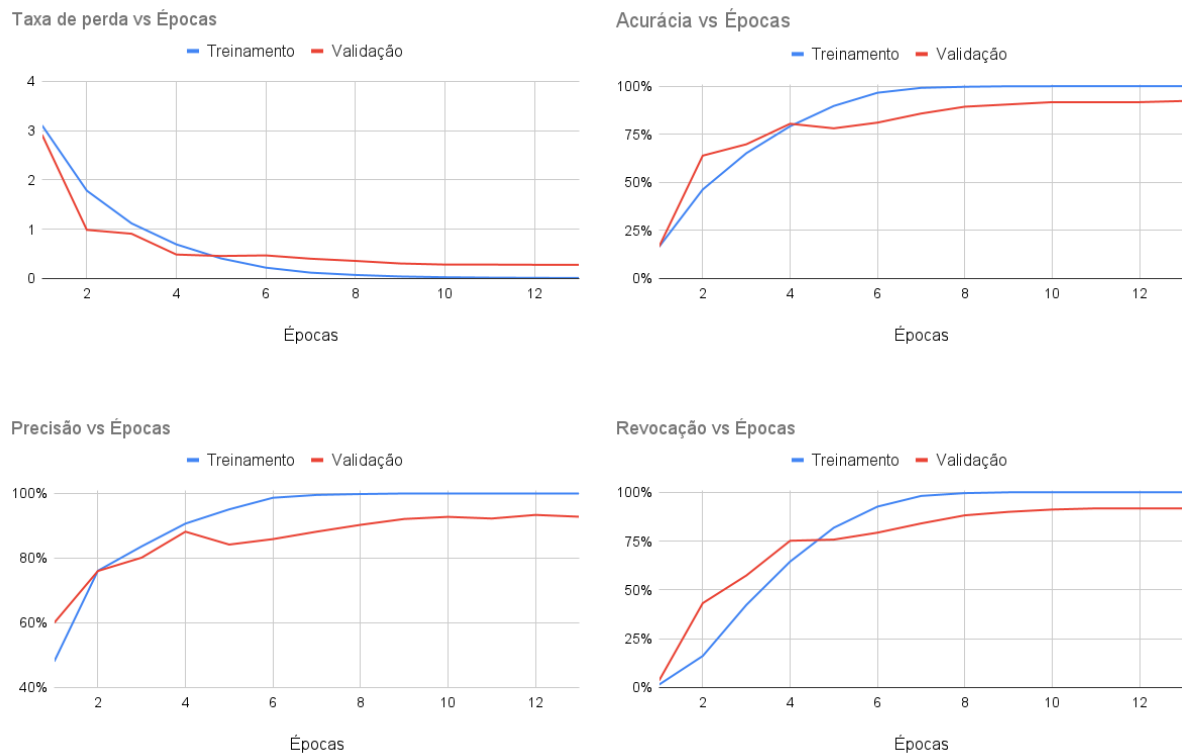
Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
37	0,87	81,15%	100,00%	51,83%	68,28%	155	36

Fonte – Danilo Assunção, 2022

No teste deste experimento foi alcançado um valor de 68,28% para a Medida F, que é a medida sendo utilizada para comparativo neste trabalho, e, considerando que este modelo não possui nenhuma técnica de aumento de dados e transferência de aprendizado, ele conseguiu atingir um valor de 81,15% de Acurácia, acertando um total de 155 imagens e errando 36 imagens.

No experimento seguinte, considerando a aplicação da técnica de aumento de dados através do uso do conjunto de dados aumentado, o modelo não mais demonstrou melhorias até seu treinamento por 13 épocas. Os dados adquiridos no treinamento deste modelo podem ser vistos na Figura 35.

Figura 35 – Resultados do treinamento da Inception v3 + Dados aumentados



Fonte – Danilo Assunção, 2022

Para a fase de teste deste modelo utilizando os dados aumentados, temos a visualização dos resultados obtidos no Quadro 6.

Quadro 6 – Resultados do teste com a Inception v3 + Dados aumentados

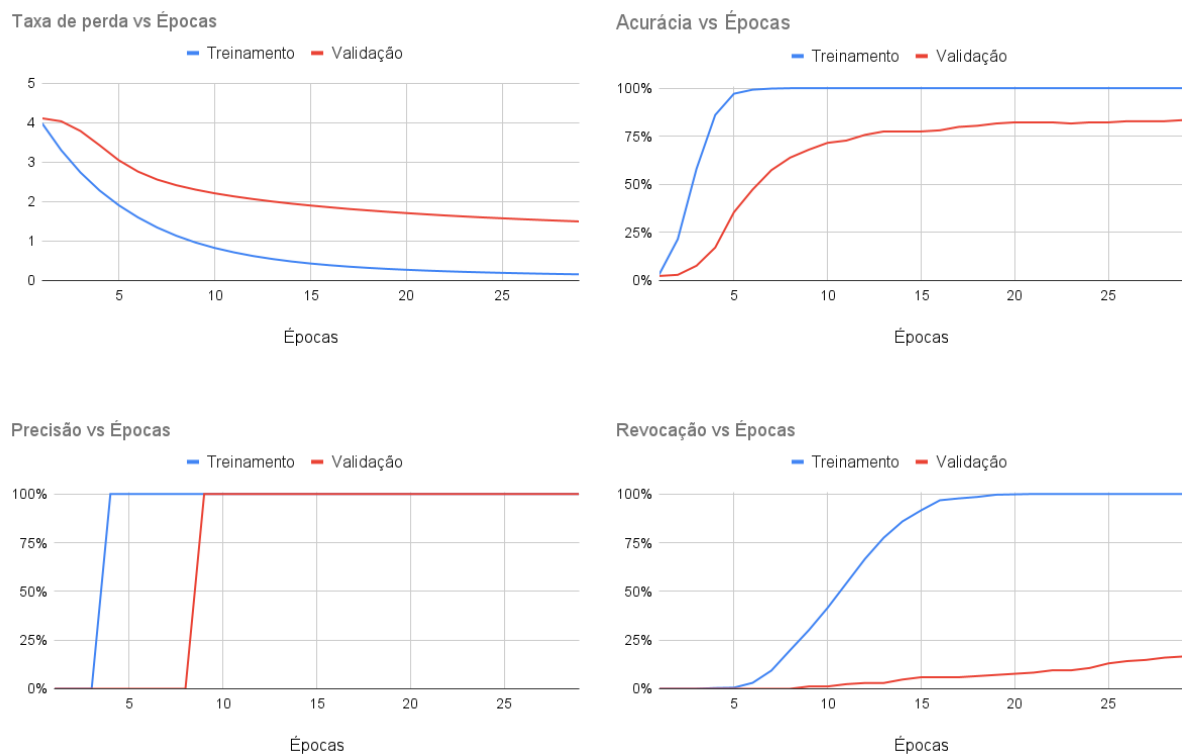
Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
13	0,28	91,10%	91,44%	89,53%	90,48%	174	17

Fonte – Danilo Assunção, 2022

Comparando a Medida F e a Acurácia resultante dos dois últimos testes, o modelo que utilizou o conjunto de dados aumentado teve um acréscimo de 22,20% em sua Medida F e 9,95% em sua Acurácia, somente ampliando a quantidade de imagens do conjunto de dados. É observável também que, além do desempenho ter sido melhorado com o aumento de dados, a quantidade de épocas necessárias para se ter este resultado foi menor em relação ao modelo que não usa o aumento de dados, tendo utilizado 24 épocas a menos para atingir resultados melhores neste modelo. Sendo assim, o uso da Inception v3 com aumento de dados, demonstrou ser superior em comparação ao experimento que utiliza o conjunto de dados original.

Considerando os modelos anteriores, foram também executados testes utilizando o conjunto de dados original e o conjunto de dados aumentado, em união com a técnica de transferência de aprendizado. O primeiro teste foi feito empregando o conjunto de dados original com a técnica de transferência de aprendizado. Seus dados obtidos no processo de treinamento podem ser observados na Figura 36.

Figura 36 – Resultados do treinamento com a Inception v3 + Dados originais + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Para o teste deste modelo foi coletado os valores que podem ser vistos através do Quadro 7.

Quadro 7 – Resultados do teste com a Inception v3 + Dados originais + Transferência de aprendizado

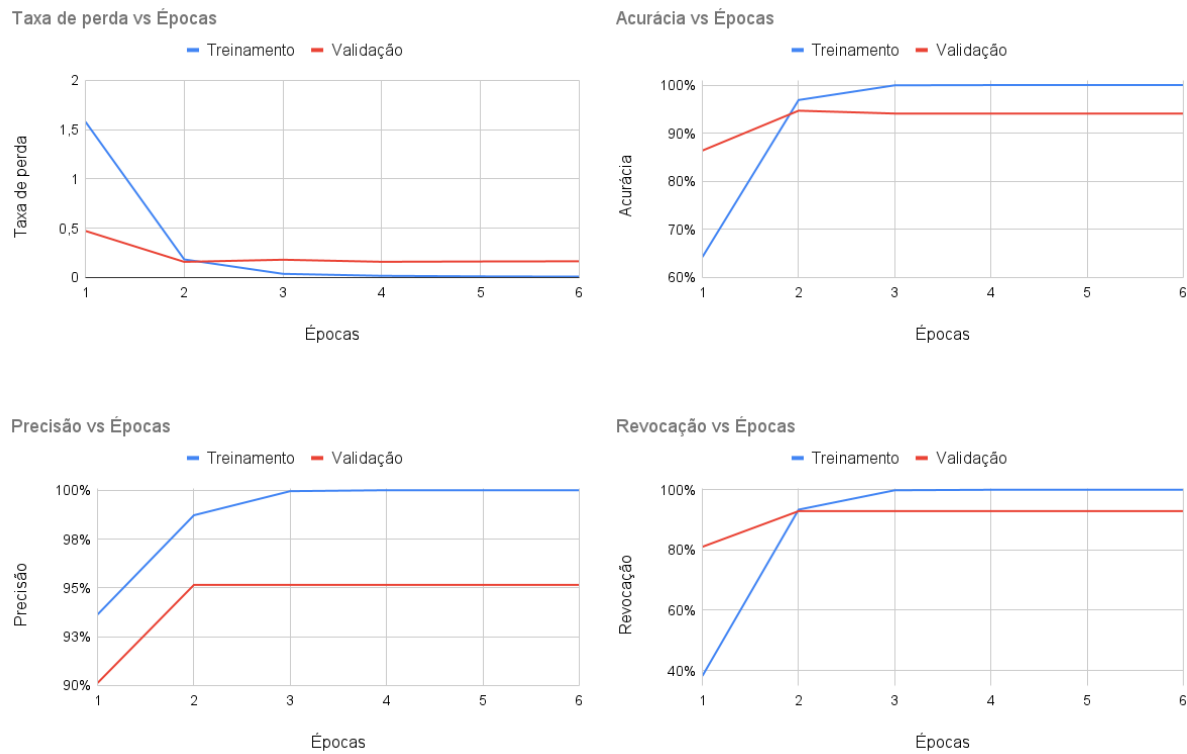
Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
37	0,87	81,15%	100,00%	51,83%	68,28%	155	36

Fonte – Danilo Assunção, 2022

Por fim, o último experimento utilizando esta arquitetura foi feito aplicando ambas as técnicas de aumento de dados e de transferência de aprendizado. Os resultados obtidos

durante o treinamento deste modelo, usando ambas as técnicas, podem ser analisados através da Figura 37.

Figura 37 – Resultados do treinamento com a Inception v3 + Dados aumentados + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Em relação ao modelo que usa transferência de aprendizado com os dados originais, este exemplo precisou de apenas 6 épocas, obtendo seu melhor resultado em 31 épocas a menos em comparação ao modelo anterior. Os dados do teste deste experimento podem ser visualizados no Quadro 8.

Quadro 8 – Resultados do teste com a Inception v3 + Dados aumentados + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
6	0,18	94,24%	95,19%	93,19%	94,18%	180	11

Fonte – Danilo Assunção, 2022

Os resultados em termos de desempenho deste último modelo, demonstraram melhores valores em comparação com a Medida F e Acurácia do experimento que utiliza a técnica de transferência de aprendizado com o conjunto de dados original, tendo obtido

uma melhoria de 25,90% em sua Medida F e 13,09% na Acurácia, utilizando 31 épocas a menos do que o modelo anterior.

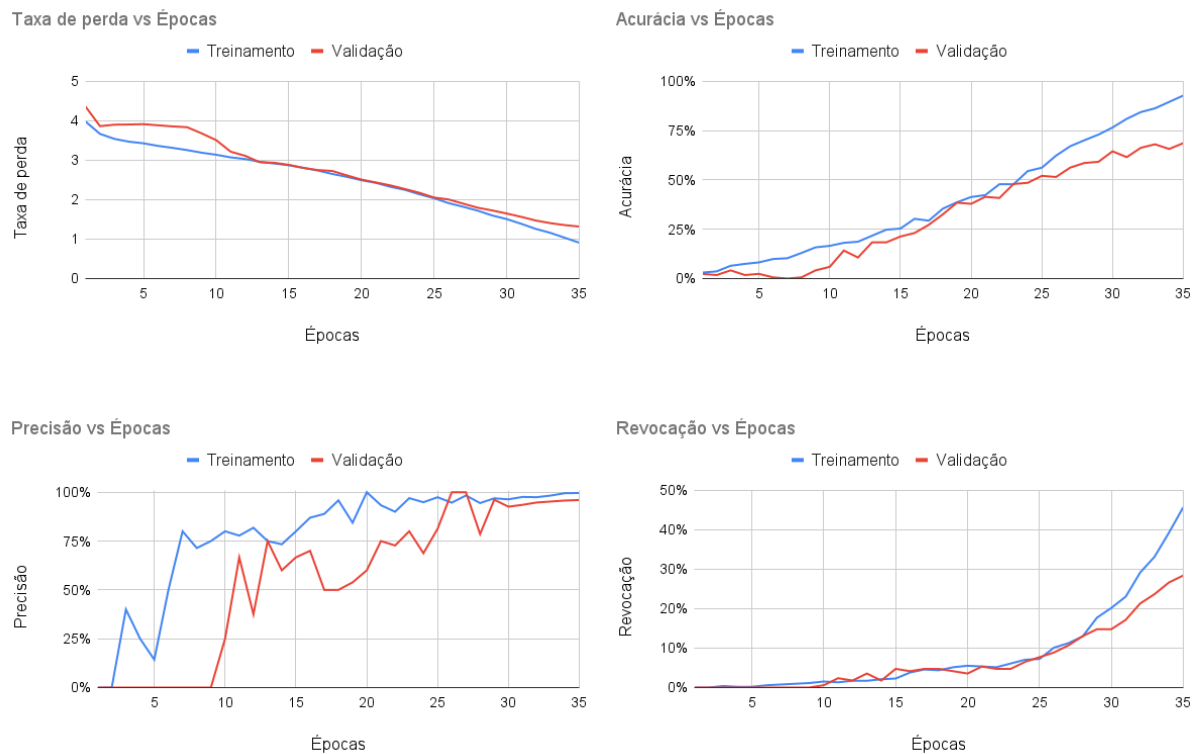
Por apresentarem os maiores resultados, foi feita uma comparação com este último experimento em relação ao experimento que utiliza apenas o conjunto de dados aumentado, em que, este último experimento obteve um desempenho melhor, atingindo um aumento de 3,70% em sua Medida F e 3,14% em sua Acurácia, treinando por 7 épocas a menos do que o modelo comparado.

5.1.3 ResNet-152

ResNet-152 é uma arquitetura de rede neural convolucional da família ResNet, possuindo uma profundidade de 152 camadas. A ResNet-152 é 8 vezes mais profunda do que as redes da família VGG, entretanto, dado a sua arquitetura, a ResNet-152 ainda possui menor complexidade (HE *et al.*, 2016) devido a certas condições presentes nesta arquitetura.

Para a arquitetura ResNet-152, o primeiro experimento realizado foi utilizando o conjunto de dados original. Este modelo foi treinado por 35 épocas até não haver mais melhorias em seus resultados. A seguir, é possível visualizar os resultados obtidos na fase de treinamento para este experimento, através da Figura 38.

Figura 38 – Resultados do treinamento com a ResNet-152 + Dados originais



Fonte – Danilo Assunção, 2022

Para este modelo, foram obtidos os resultados e coletadas as suas medidas, sendo elas apresentadas no Quadro 9.

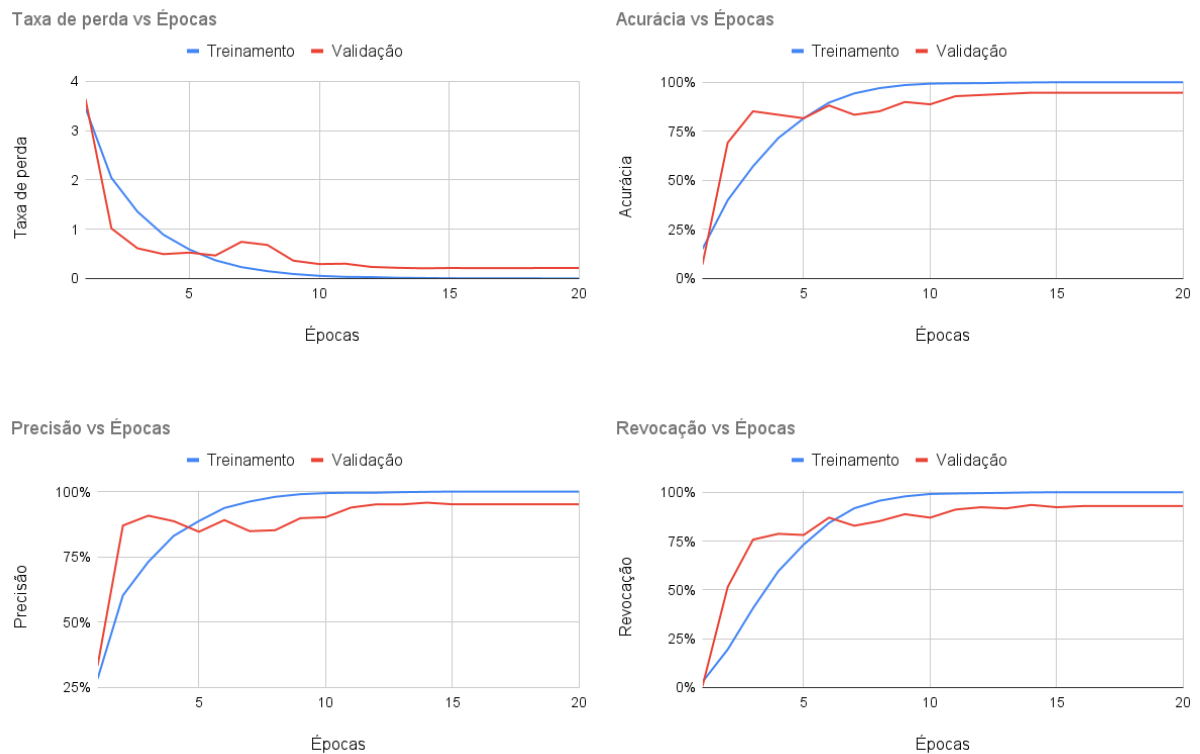
Quadro 9 – Resultados do teste com a ResNet-152 + Dados originais

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
35	1,34	69,11%	90,32%	29,32%	44,27%	132	59

Fonte – Danilo Assunção, 2022

No experimento seguinte, foi feita a aplicação da técnica de aumento de dados através do uso do conjunto de dados aumentado, em que o modelo foi treinado por 20 épocas até não demonstrar mais melhorias em seu desempenho. Os dados adquiridos no treinamento deste modelo podem ser vistos na Figura 39.

Figura 39 – Resultados do treinamento da ResNet-152 + Dados aumentados



Fonte – Danilo Assunção, 2022

Para a fase de teste deste modelo utilizando os dados aumentados, temos a visualização dos resultados obtidos no Quadro 10.

Quadro 10 – Resultados do teste com a ResNet-152 + Dados aumentados

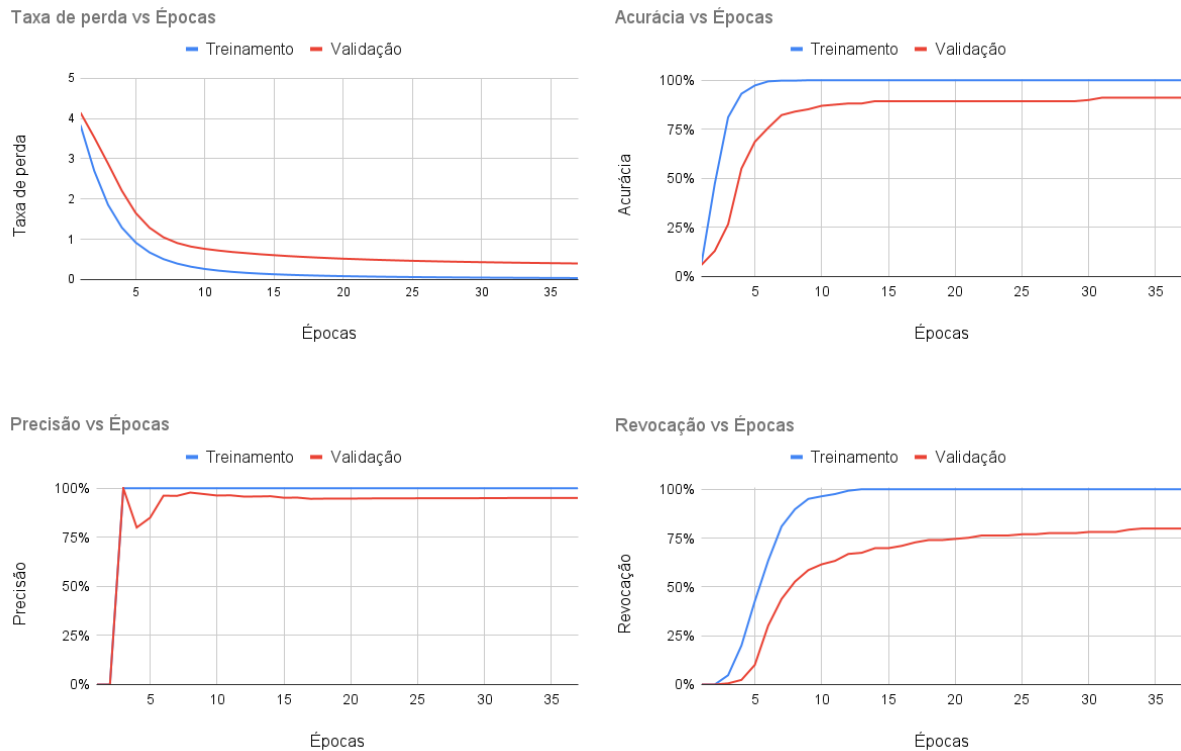
Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
20	0,22	93,19%	94,18%	93,19%	93,68%	178	13

Fonte – Danilo Assunção, 2022

Comparando a Medida F e a Acurácia resultante dos dois últimos testes, o modelo que utilizou o conjunto de dados aumentado teve um acréscimo de 49,41% em sua Medida F e 24,08% em sua Acurácia, somente ampliando a quantidade de imagens do conjunto de dados. Contudo, além do desempenho ter sido melhorado com o aumento de dados, a quantidade de épocas necessárias para se ter este resultado foi menor em relação ao modelo que usa o conjunto de dados original, tendo utilizado 15 épocas a menos para se atingir estes resultados.

O teste a seguir foi feito empregando o conjunto de dados original com a técnica de transferência de aprendizado. Seus dados obtidos no processo de treinamento podem ser observados na Figura 40.

Figura 40 – Resultados do treinamento com a ResNet-152 + Dados originais + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Para o teste deste modelo foi coletado os valores que podem ser vistos através do Quadro 11.

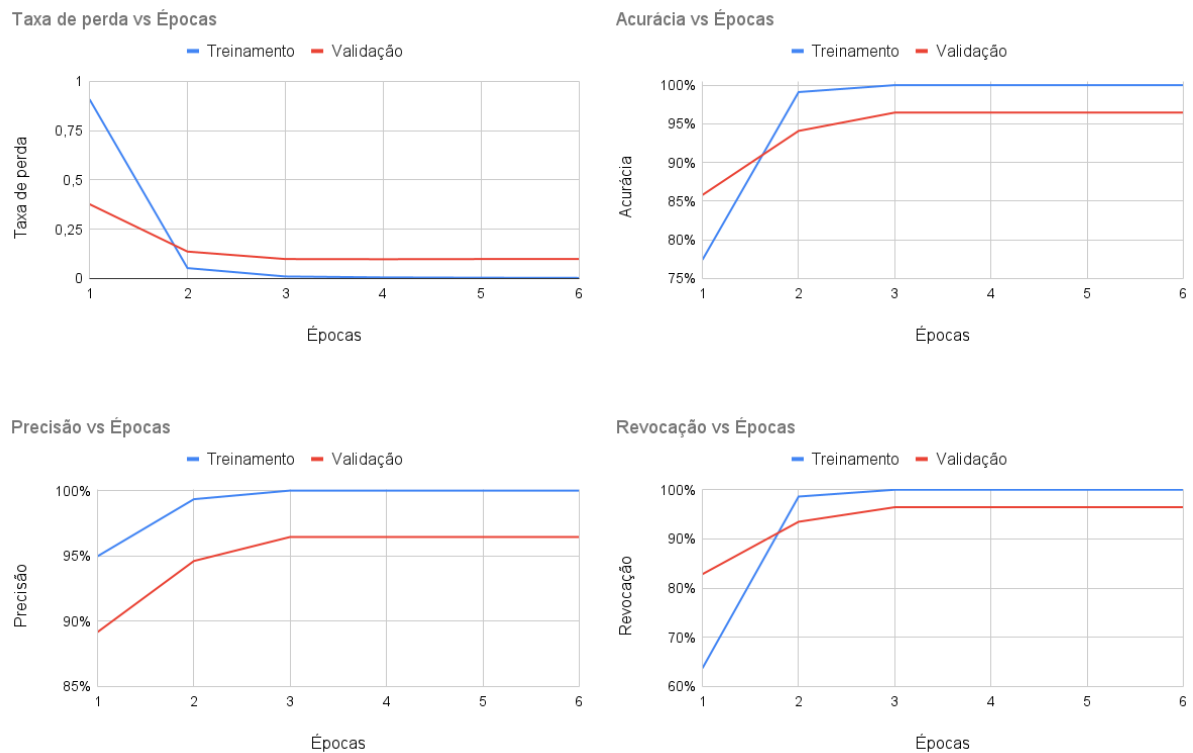
Quadro 11 – Resultados do teste com a ResNet-152 + Dados originais + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
37	0,40	93,72%	96,91%	82,20%	88,95%	179	12

Fonte – Danilo Assunção, 2022

Por fim, o último experimento utilizando esta arquitetura foi feito aplicando ambas as técnicas de aumento de dados e de transferência de aprendizado. Os resultados obtidos durante o treinamento deste modelo, usando ambas as técnicas, podem ser analisados através da Figura 41.

Figura 41 – Resultados do treinamento com a ResNet-152 + Dados aumentados + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Os dados do teste deste experimento podem ser visualizados no Quadro 12.

Quadro 12 – Resultados do teste com a ResNet-152 + Dados aumentados + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
6	0,10	96,34%	96,34%	96,34%	96,34%	184	7

Fonte – Danilo Assunção, 2022

Os resultados em termos de desempenho deste último modelo, demonstraram melhores valores em comparação com a Medida F e Acurácia do experimento que utiliza a técnica de transferência de aprendizado com o conjunto de dados original, tendo obtido uma melhoria de 7,39% em sua Medida F e 2,62% na Acurácia, utilizando 31 épocas a menos do que o modelo anterior.

Uma comparação foi realizada a partir dos dois experimentos que trouxeram os maiores resultados, sendo eles o último experimento e o experimento que utiliza apenas o conjunto de dados aumentado. O último experimento acabou por ter um desempenho melhor ao atingir um aumento de 2,66% em sua Medida F e 3,15% em sua Acurácia,

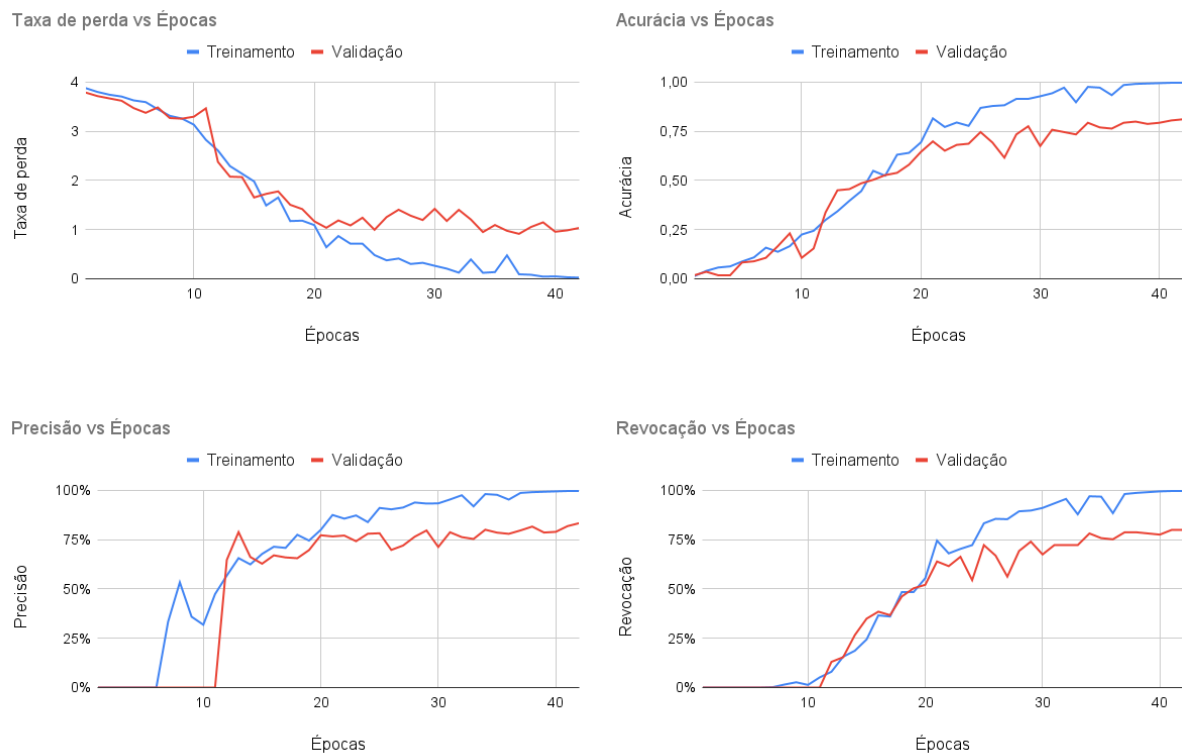
apresentando esse resultado ao treinar por 14 épocas a menos do que o modelo comparado e utilizando ambas as técnicas.

5.1.4 VGG16

VGG é uma arquitetura de CNN clássica, baseada em uma análise de como aumentar a profundidade de tais redes, proposta em [Simonyan e Zisserman \(2014\)](#). A VGG16 é uma arquitetura que faz parte da família de arquiteturas VGG contendo um total de 16 camadas.

Para a arquitetura VGG16, o primeiro experimento realizado foi utilizando o conjunto de dados original. Este modelo foi treinado por 42 épocas até não haver mais melhorias em seus resultados. A seguir, é possível visualizar os resultados obtidos na fase de treinamento para este experimento, através da [Figura 42](#).

Figura 42 – Resultados do treinamento com a VGG16 + Dados originais



Fonte – Danilo Assunção, 2022

Para este modelo, foram obtidos os resultados e coletadas as suas medidas, sendo elas apresentadas no [Quadro 13](#).

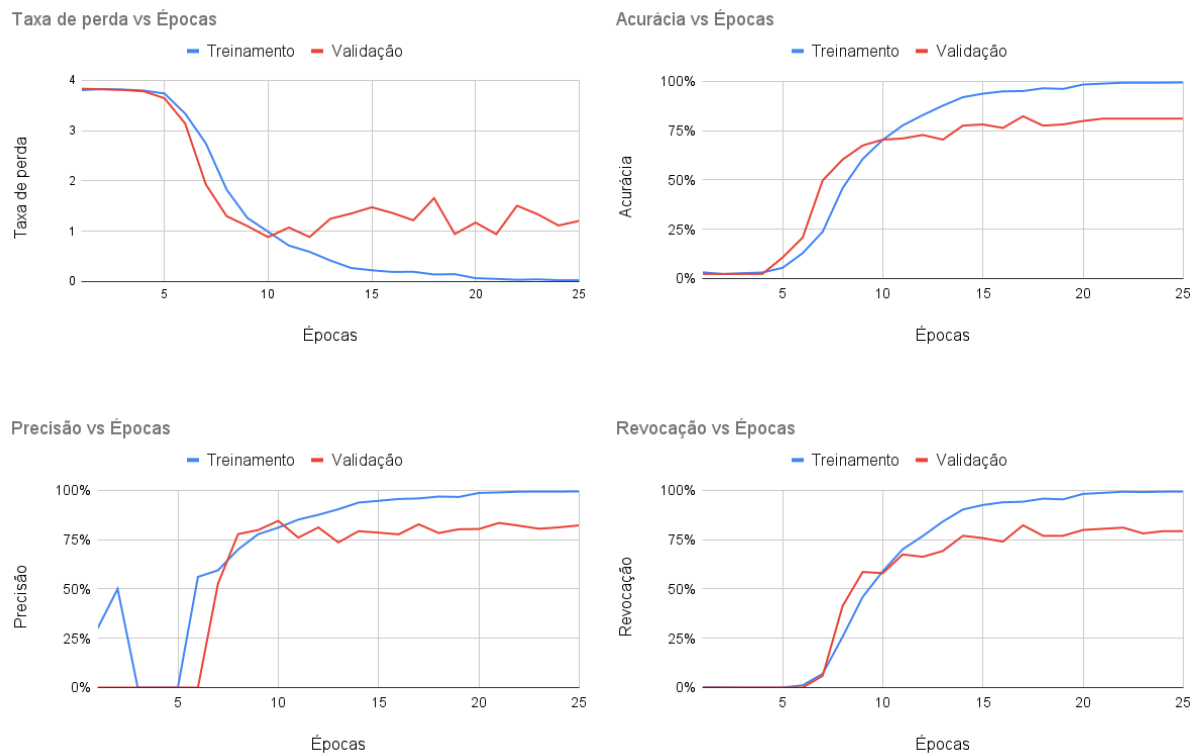
Quadro 13 – Resultados do teste com a VGG16 + Dados originais

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
42	1,05	79,06%	81,87%	78,01%	79,89%	151	40

Fonte – Danilo Assunção, 2022

No experimento seguinte, foi feita a aplicação da técnica de aumento de dados através do uso do conjunto de dados aumentado, com este modelo sendo treinado por um total de 25 épocas até não demonstrar mais melhorias em seu desempenho. Os dados adquiridos no treinamento deste modelo podem ser vistos na Figura 43.

Figura 43 – Resultados do treinamento da VGG16 + Dados aumentados



Fonte – Danilo Assunção, 2022

Para a fase de teste deste modelo utilizando os dados aumentados, temos a visualização dos resultados obtidos no Quadro 14.

Quadro 14 – Resultados do teste com a VGG16 + Dados aumentados

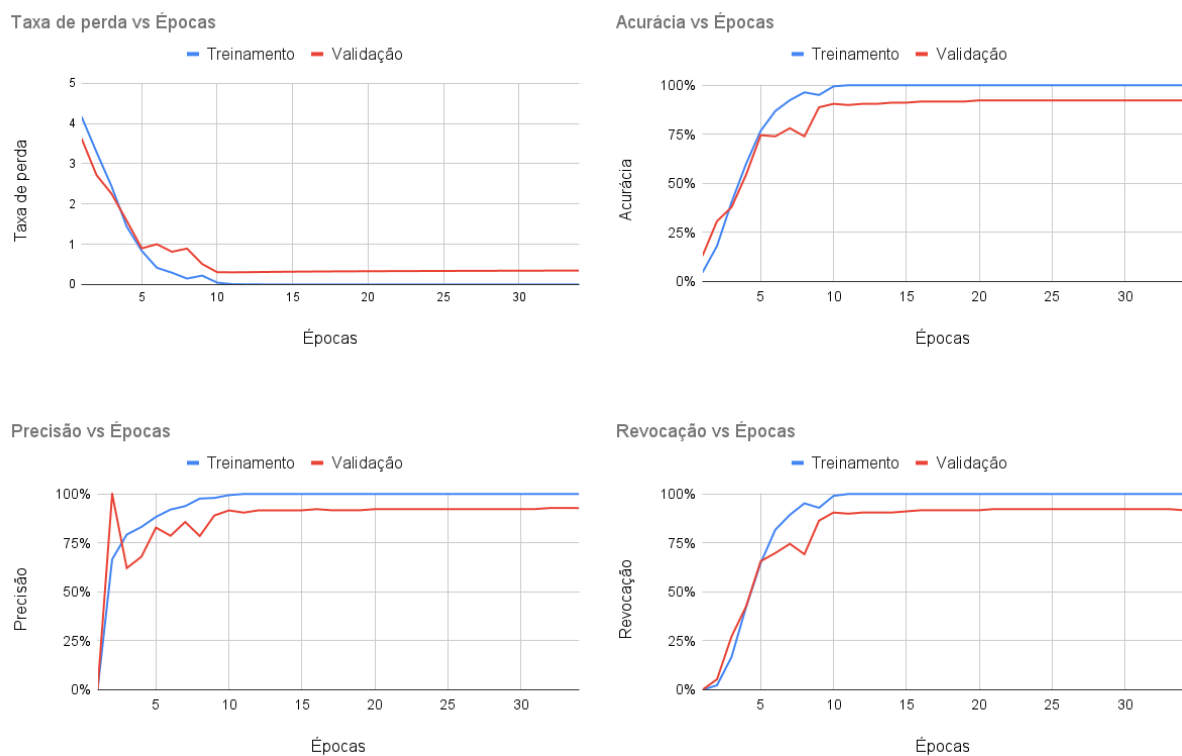
Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
25	1,33	77,49%	79,35%	76,44%	77,87%	148	43

Fonte – Danilo Assunção, 2022

Comparando a Medida F e a Acurácia resultante dos dois últimos testes, o modelo que utilizou o conjunto de dados original ainda teve um maior desempenho com relação ao experimento que usa conjunto de dados aumentado, tendo uma diferença de 2,02% na Medida F e 1,57% na Acurácia. Este foi o primeiro experimento no qual a técnica de aumento de dados não demonstrou melhores resultados com relação ao experimento usando apenas o conjunto de dados original. Entretanto, ainda houve uma diferença de 17 épocas para que o experimento que usa os dados originais obtivesse tais resultados.

O teste a seguir foi feito aplicando o conjunto de dados original com a técnica de transferência de aprendizado. Os dados obtidos no processo de treinamento deste experimento podem ser vistos na Figura 44.

Figura 44 – Resultados do treinamento com a VGG16 + Dados originais + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Para o teste deste modelo foi coletado os valores que podem ser vistos através do Quadro 15.

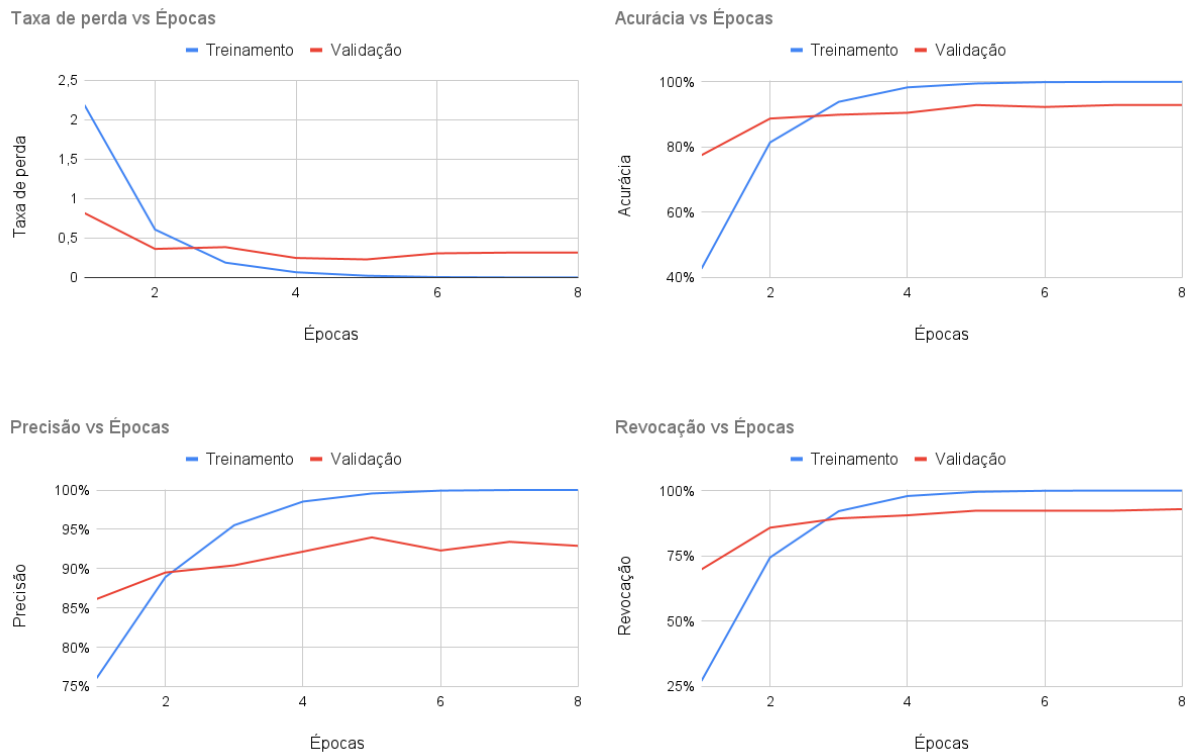
Quadro 15 – Resultados do teste com a VGG16 + Dados originais + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
34	0,31	90,58%	90,58%	90,58%	90,58%	173	18

Fonte – Danilo Assunção, 2022

E por último, foi feito o teste aplicando as técnicas de aumento de dados e de transferência de aprendizado. Os resultados obtidos durante o treinamento deste modelo, usando ambas as técnicas, podem ser analisados através da Figura 45.

Figura 45 – Resultados do treinamento com a VGG16 + Dados aumentados + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Os dados do teste deste experimento podem ser visualizados no Quadro 16.

Quadro 16 – Resultados do teste com a VGG16 + Dados aumentados + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
8	0,26	93,19%	93,19%	93,19%	93,19%	178	13

Fonte – Danilo Assunção, 2022

Os resultados em termos de desempenho deste último modelo, demonstraram melhores valores em comparação com a Medida F e Acurácia do experimento que utiliza a técnica de transferência de aprendizado com o conjunto de dados original, tendo obtido uma melhoria de 2,61% em sua Medida F e 2,61% na Acurácia e utilizando 26 épocas a menos do que o modelo anterior.

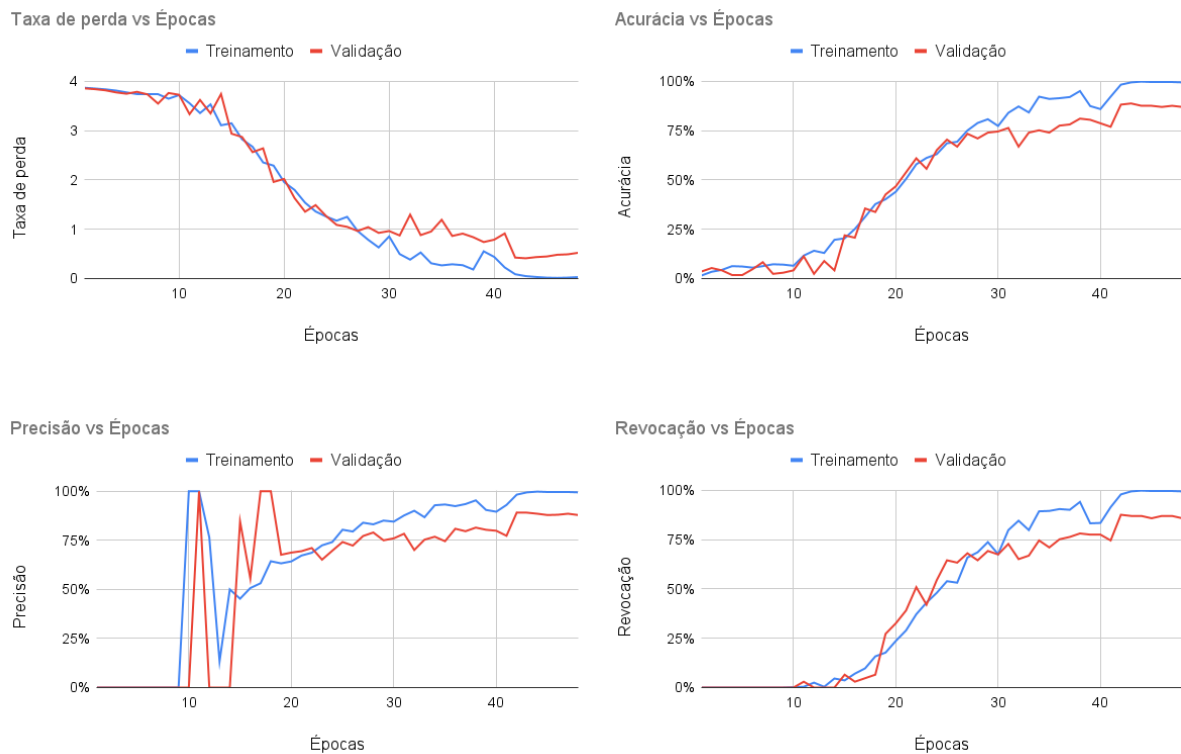
Neste experimento utilizando a arquitetura VGG16, os dois modelos que melhor desempenharam foram ambos os exemplares que utilizam a técnica de transferência de aprendizado, sendo assim, para esta arquitetura, a técnica de transferência de aprendizado foi a que melhor auxiliou em termos de desempenho.

5.1.5 VGG19

VGG19 é uma arquitetura de CNN que pertence à família VGG, possuindo um total de 19 camadas que compõem a sua estrutura.

Para a arquitetura VGG19, o primeiro experimento realizado foi utilizando o conjunto de dados original. Este modelo foi treinado por 48 épocas até não apresentar mais melhorias em seus resultados. A seguir, é possível visualizar os resultados obtidos na fase de treinamento para este experimento, através da Figura 46.

Figura 46 – Resultados do treinamento com a VGG19 + Dados originais



Fonte – Danilo Assunção, 2022

Para este modelo, foram obtidos os resultados e coletadas as suas medidas, sendo elas apresentadas no Quadro 17.

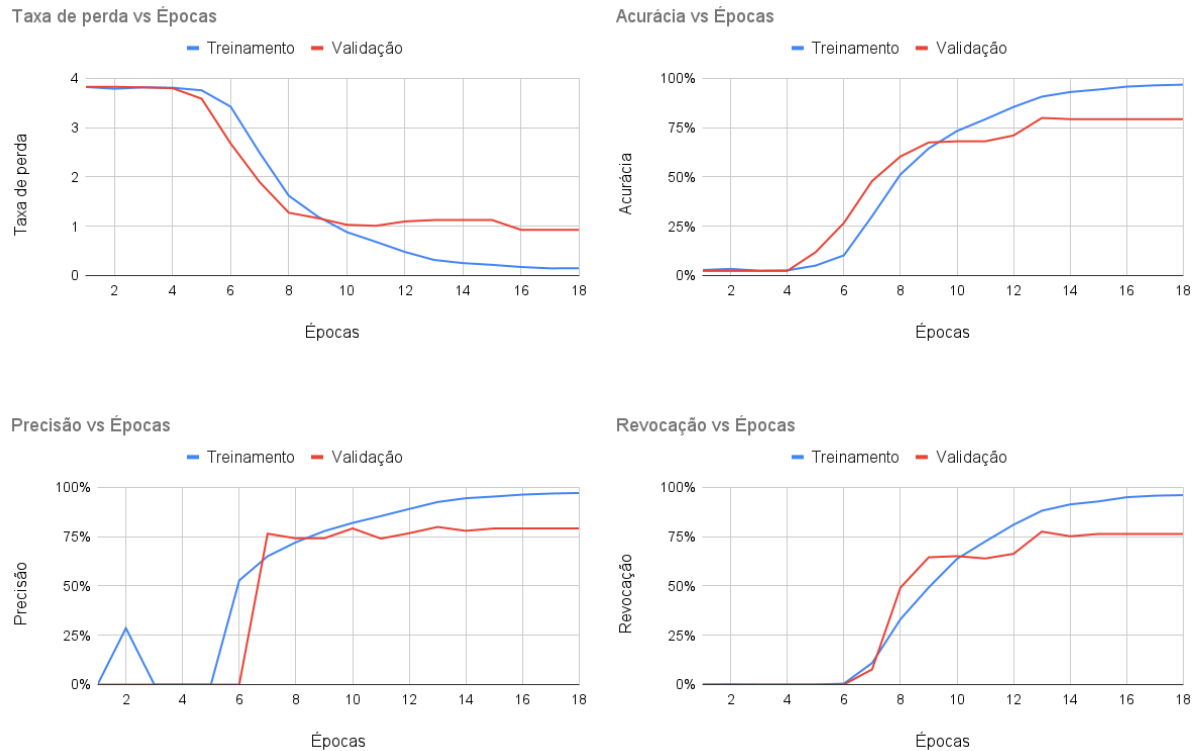
Quadro 17 – Resultados do teste com a VGG19 + Dados originais

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
48	0,84	82,72%	84,41%	82,20%	83,29%	158	33

Fonte – Danilo Assunção, 2022

No experimento seguinte, foi feita a aplicação da técnica de aumento de dados através do uso do conjunto de dados aumentado, em que o modelo foi treinado por um total de 18 épocas até não demonstrar mais melhorias em seu desempenho. Os dados adquiridos no treinamento deste modelo podem ser vistos na Figura 47.

Figura 47 – Resultados do treinamento da VGG19 + Dados aumentados



Fonte – Danilo Assunção, 2022

Para a fase de teste deste modelo utilizando os dados aumentados, temos a visualização dos resultados obtidos no Quadro 18.

Quadro 18 – Resultados do teste com a VGG19 + Dados aumentados

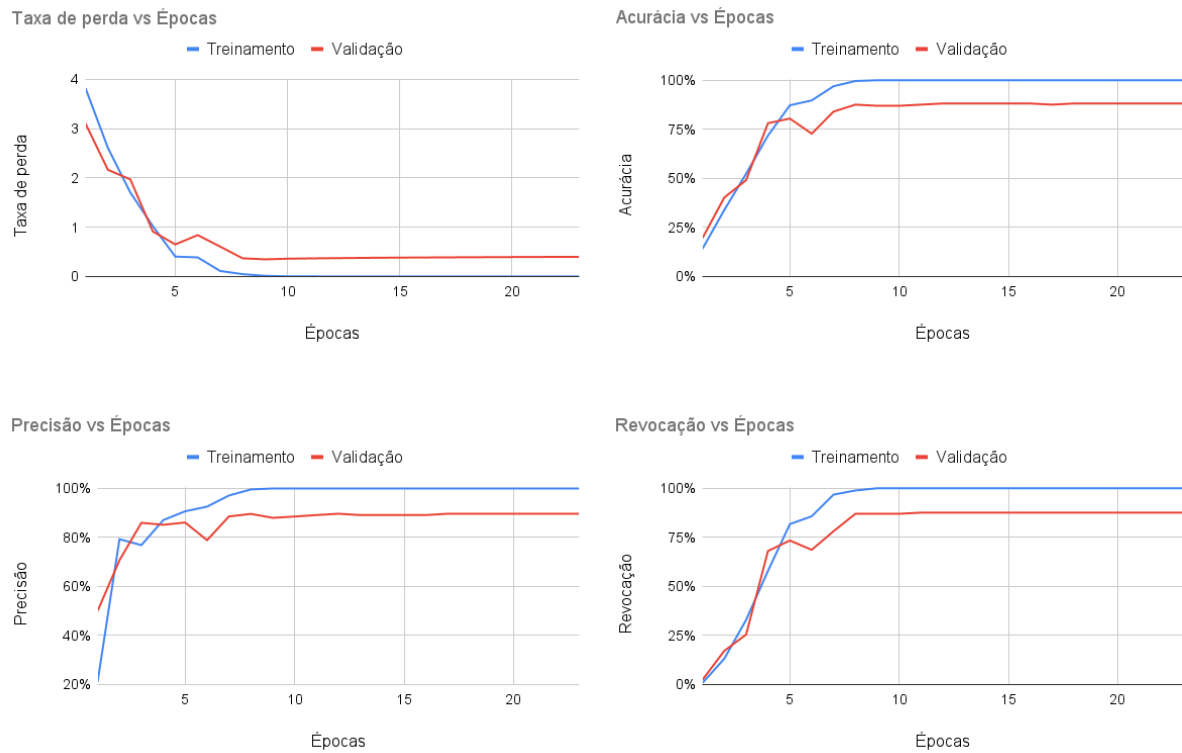
Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
18	1,47	80,10%	79,89%	79,06%	79,47%	153	38

Fonte – Danilo Assunção, 2022

Comparando a Medida F e a Acurácia resultante dos dois últimos testes, o modelo que utilizou o conjunto de dados original ainda teve um maior desempenho em relação ao experimento que usa conjunto de dados aumentado, tendo uma diferença de 3,82% na Medida F e 2,62% na Acurácia, tendo percorrido 30 épocas a mais para se atingir estes resultados. Este foi um resultado similar ao cenário visto na VGG16, onde o modelo que usa o conjunto de dados aumentado obteve um desempenho menor em relação ao modelo que usa o conjunto de dados original.

O teste a seguir foi feito aplicando o conjunto de dados original com a técnica de transferência de aprendizado. Os dados obtidos no processo de treinamento deste experimento podem ser vistos na Figura 48.

Figura 48 – Resultados do treinamento com a VGG19 + Dados originais + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Para o teste deste modelo foi coletado os valores que podem ser vistos através do Quadro 19.

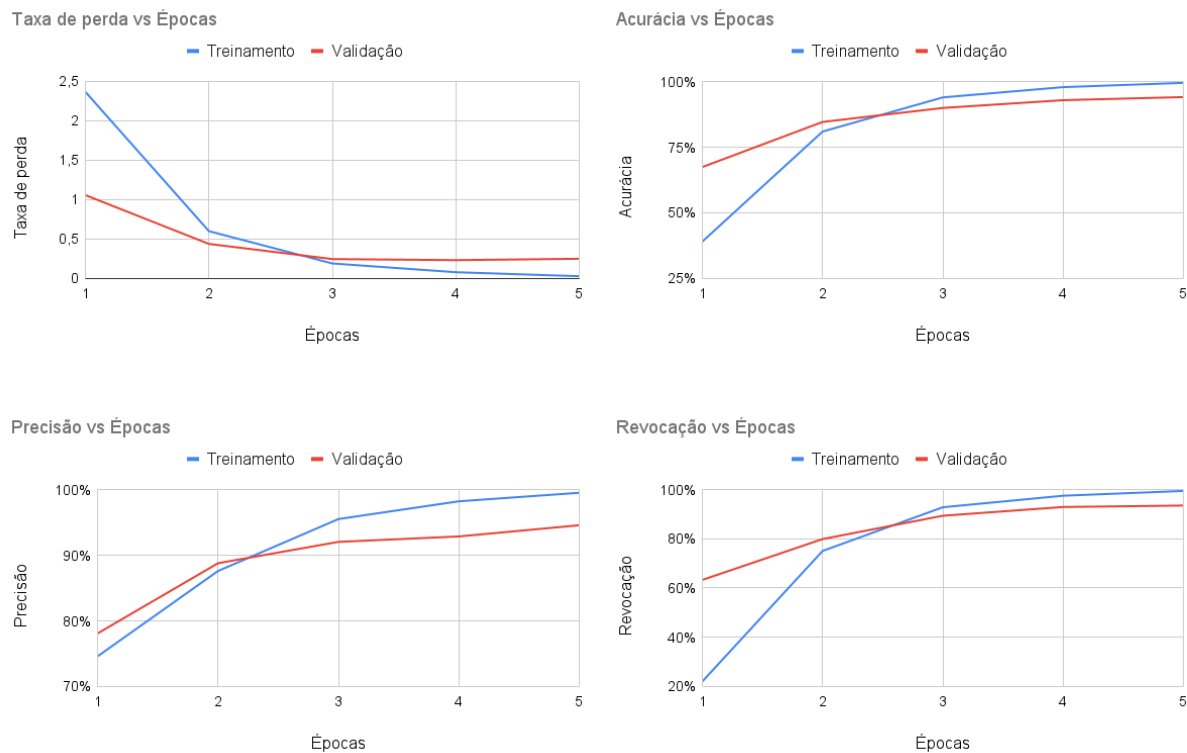
Quadro 19 – Resultados do teste com a VGG19 + Dados originais + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
23	0,51	89,01%	90,32%	87,96%	89,12%	170	21

Fonte – Danilo Assunção, 2022

Por fim, foi executado o teste aplicando ambas as técnicas de aumento de dados e transferência de aprendizado. Os resultados obtidos durante o treinamento deste modelo podem ser analisados através da Figura 49.

Figura 49 – Resultados do treinamento com a VGG19 + Dados aumentados + Transferência de aprendizado



Fonte – Danilo Assunção, 2022

Os dados do teste deste experimento podem ser visualizados no Quadro 20.

Quadro 20 – Resultados do teste com a VGG19 + Dados aumentados + Transferência de aprendizado

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
5	0,27	94,24%	94,74%	94,24%	94,49%	180	11

Fonte – Danilo Assunção, 2022

Os resultados em termos de desempenho deste último modelo, demonstraram melhores valores em comparação com a Medida F e Acurácia do experimento que utiliza a técnica de transferência de aprendizado com o conjunto de dados original, tendo obtido uma diferença superior de 5,37% em sua Medida F e 5,23% em na Acurácia utilizando 18 épocas a menos do que o modelo anterior.

Neste experimento utilizando a arquitetura VGG19, os dois modelos que melhor desempenharam foram também os exemplares que utilizam a técnica de transferência de aprendizado, sendo assim, para esta arquitetura, a técnica de transferência de aprendizado

foi a que também melhor auxiliou em termos de desempenho para os modelos que usaram a arquitetura VGG.

5.1.6 Conclusão

Todas as arquiteturas de CNN utilizadas nos experimentos desta seção possuem suas especificidades, isto é, cada uma contém características que permitem a resolução de um determinado problema. Se tratando das técnicas de aumento de dados e transferência de aprendizado, a técnica que mais influenciou em termos de desempenho foi a técnica de aumento de dados, que proporcionou resultados relevantes para os modelos em comparação aos modelos que utilizaram o conjunto de dados original, entretanto, a técnica de transferência de aprendizado também teve uma participação no aspecto da quantidade de épocas nas quais esses modelos foram treinados, dado que, com o uso desta técnica, foi notado uma considerável queda na quantidade de épocas necessárias para se atingir certos resultados, e, com a união de ambas as técnicas, têm-se normalmente os mesmos comportamentos presentes ao mesmo tempo, elevando consideravelmente o resultado obtido nos testes, já que as medidas resultantes do modelo eram melhoradas e o número de épocas necessário para se atingir estes resultados eram menores.

Um outro aspecto também observado através dos gráficos contendo as medidas obtidas na fase de treinamento dos experimentos, é a incidência de cenários com sobreajuste (Figura 36) ou subajuste, (Figura 38) sendo tais cenários predominantemente encontrados apenas em modelos que utilizaram o conjunto de dados original, e dado este comportamento, foi possível identificar que as técnicas de aumento de dados e transparência ajudaram muito a mitigar casos de sobreajuste ou subajuste nos modelos que utilizam estes métodos, em principal a técnica de aumento de dados, tornando-se um bom indicativo que, para a prevenção de sobreajuste ou subajuste, o uso de técnicas de aumento de dados e transferência de aprendizado podem favorecer contra este tipo de problema, evitando que cenários deste tipo ocorram.

Concluindo, neste experimento o modelo que melhor desempenhou em termos de Acurácia e Medida F foi a arquitetura ResNet-152, que fez uso das técnicas de aumento de dados e transferência de aprendizado, e retornou o melhor resultado em relação aos outros modelos ao obter uma Acurácia e Medida F de 96,33% respectivamente.

5.2 Seleção do algoritmo de otimização

Nesta seção serão abordadas algumas experimentações com base na arquitetura de CNN extraída no primeiro experimento, isto é, utilizando-se da arquitetura de CNN selecionada na primeira experimentação (ResNet-152) e realizados outros testes em cima desta arquitetura, aplicando algoritmos de otimização distintos para verificar qual o algoritmo que possui o melhor resultado, em referência ao modelo proposto. O conjunto de dados considerado neste segundo experimento foi o conjunto de dados ampliado pela técnica de aumento de dados, visto que, a arquitetura selecionada no primeiro experimento obteve o melhor resultado utilizando o conjunto de dados aumentado. Contudo, para esta experimentação, foi feito o uso de alguns algoritmos de otimização para CNNs disponibilizados pela biblioteca utilizada nesta ferramenta, sendo estes algoritmos, SGD, Adagrad, Adadelta e Adamax. O detalhamento dos resultados obtidos do SGD já foram esclarecidos na Subseção 5.1.3 a partir da Figura 41, sendo que no primeiro experimento, as CNNs utilizam por padrão o algoritmo de otimização SGD, e o resultado obtido neste experimento para as medidas, tanto a Acurácia quanto a Medida F, foi de 96,33%. Para todos estes algoritmos, nenhum hiperparâmetro é alterado, além da estratégia que determina a taxa de aprendizado, definida na Seção 4.3, ou seja, desconsiderando a taxa de aprendizado, o restante dos hiperparâmetros são os já definidos pela biblioteca utilizada pela ferramenta.

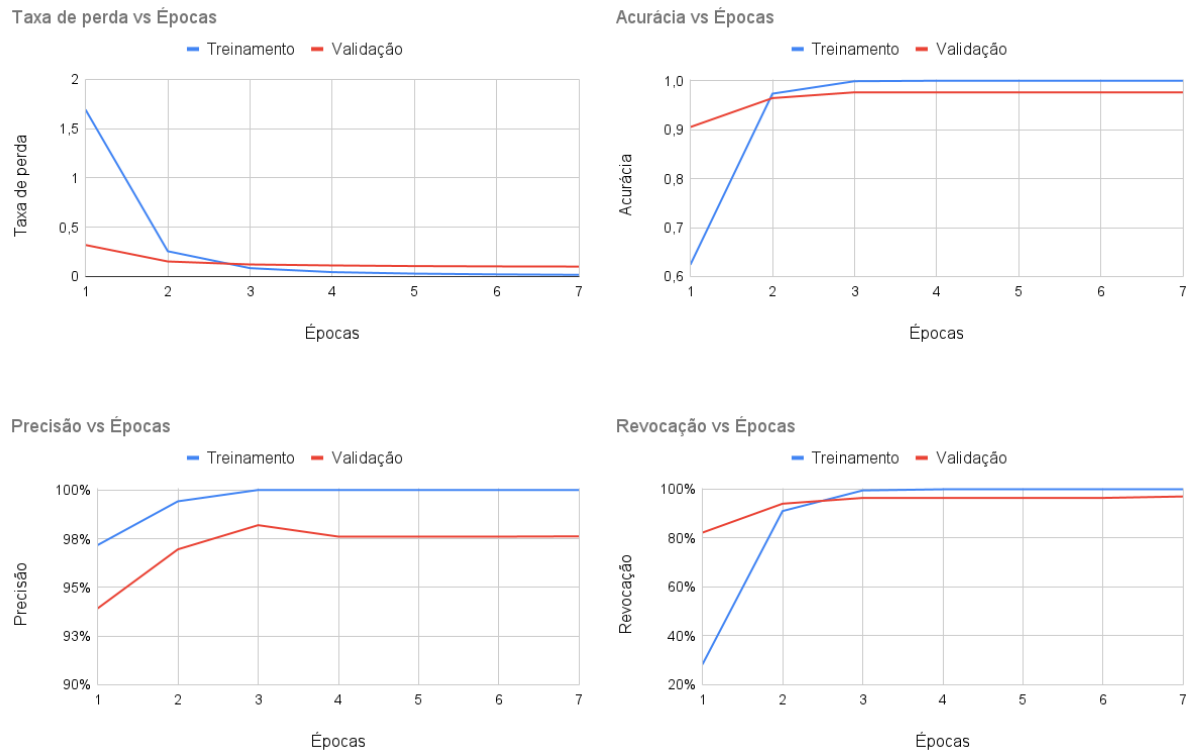
5.2.1 Adagrad

Adagrad é um algoritmo de otimização definido em [Duchi, Hazan e Singer \(2011\)](#), que aplica taxas de aprendizado específicas para os parâmetros da rede, que são adaptadas em relação à frequência com que um parâmetro é atualizado durante o treinamento. Quanto mais atualizações um parâmetro recebe, menores são as atualizações.

O primeiro passo deste teste foi feito executando o modelo selecionado no primeiro experimento que definia a arquitetura de CNN junto com o otimizador Adagrad. O treinamento deste modelo foi parado a partir da época 7, dado que não houve melhorias após esta época considerando os critérios de parada estabelecidos para o treinamento dos

modelos em todos os experimentos. Os dados obtidos no treinamento deste modelo são mostrados através da Figura 50.

Figura 50 – Resultados do treinamento usando Adagrad



Fonte – Danilo Assunção, 2022

Abaixo é exibido a partir do Quadro 21 os dados obtidos no teste deste modelo depois da fase de treinamento.

Quadro 21 – Resultados do teste usando

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
7	0,13	94,76%	95,74%	94,24%	94,99%	181	10

Fonte – Danilo Assunção, 2022

Considerando os resultados obtidos, foram de fato relevantes, porém se comparado com o modelo utilizando SGD, o Adagrad não desempenhou tão bem e isso pode ter ligação com a taxa de aprendizado utilizada na inicialização do algoritmo, dado que o valor foi estabelecido seguindo os valores já definidos para a avaliação deste trabalho. Comparando este modelo que usou Adagrad com o modelo que utilizou o otimizador SGD, é apresentado no teste, usando o Adagrad, uma queda na Medida F e Acurácia de 1,34% e 1,57% , respectivamente com 1 época a mais executada.

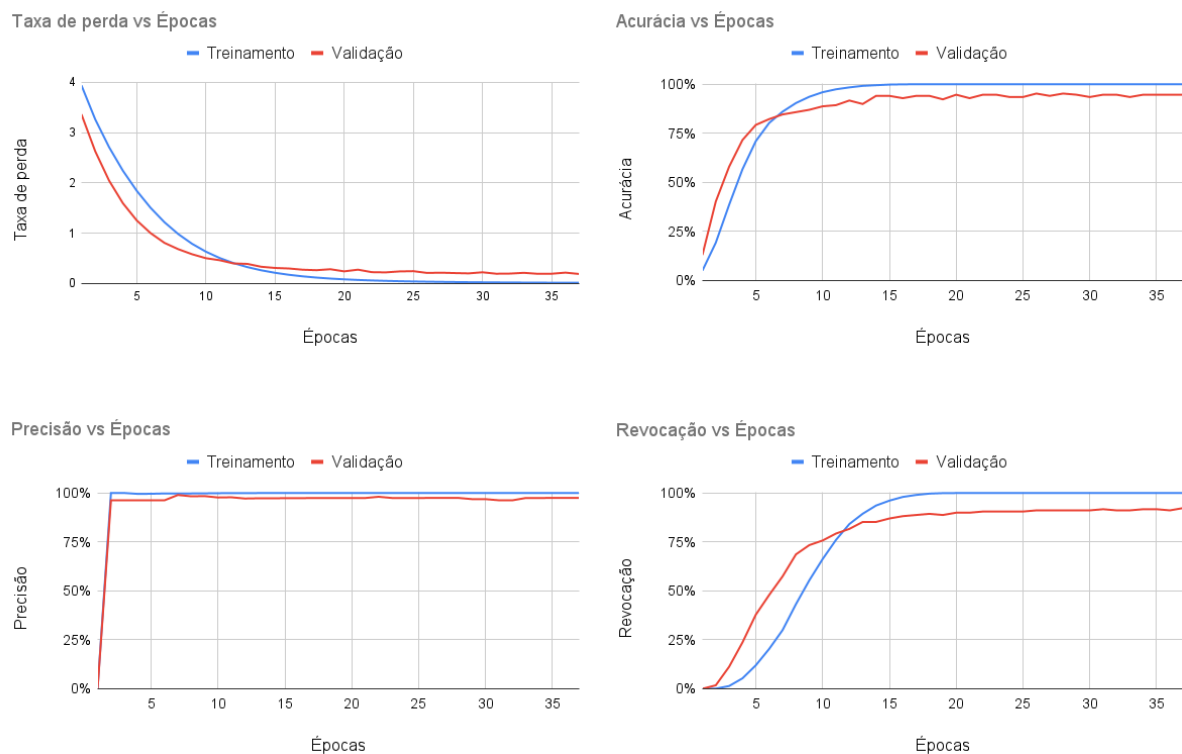
5.2.2 Adadelta

Com base na definição em Zeiler (2012), Adadelta é um método de descida de gradiente (otimizador), que se baseia na taxa de aprendizagem adaptativa por dimensão para resolver duas desvantagens contidas em otimizadores como o Adagrad, sendo elas, o decaimento contínuo das taxas de aprendizado ao longo do treinamento que se torna infinitesimalmente pequena em algum ponto, dado que a partir de um certo número de iterações o modelo pode perder a capacidade de aprender novos padrões, e, respectivamente, a necessidade da definição manual de uma taxa de aprendizado global.

O primeiro passo deste teste foi feito executando o modelo selecionado no primeiro experimento que definia a arquitetura de CNN junto com o otimizador Adadelta. O treinamento deste modelo foi parado a partir da época 37, dado que não houve melhorias após esta época considerando os critérios de parada estabelecidos para o treinamento dos modelos em todos os experimentos. Os dados obtidos no treinamento deste modelo são mostrados através da Figura 50.

Na imagem 51

Figura 51 – Resultados do treinamento usando Adadelta



Abaixo é exibido, a partir do Quadro 22, os dados obtidos no teste deste modelo depois da fase de treinamento.

Quadro 22 – Resultados do teste usando Adadelta

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
37	0,14	94,24%	95,65%	92,15%	93,87%	180	11

Fonte – Danilo Assunção, 2022

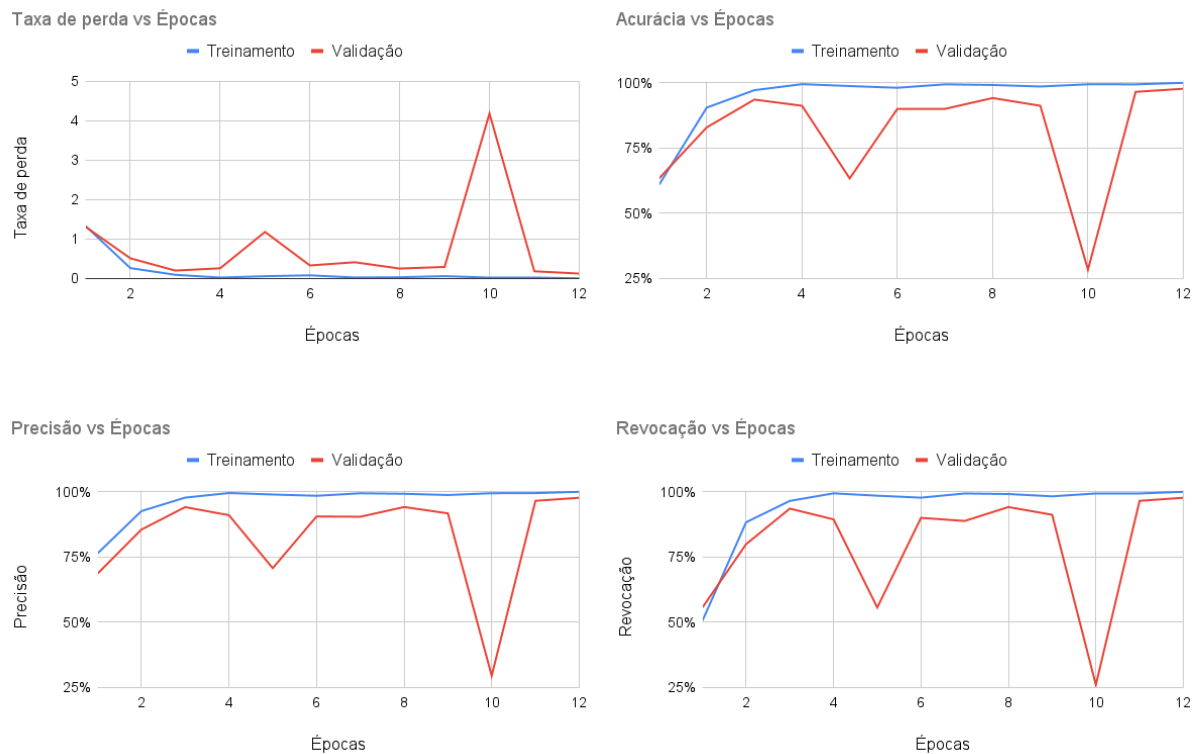
Considerando os resultados obtidos, foram de fato relevantes, porém se comparado com o modelo utilizando SGD, o Adadelta teve um desempenho levemente inferior, isto é, comparando ambos os testes destes modelos, o que usou o Adadelta teve uma diferença abaixo do teste usando SGD de 2,46% e 2,09% na Medida F e Acurácia, respectivamente com 31 épocas a mais em ciclos de treinamento.

5.2.3 Adamax

Adamax é um algoritmo de otimização adaptativo sendo uma variante do algoritmo Adam, baseado na norma do infinito. Ambos foram detalhados em [Kingma e Ba \(2014\)](#).

O primeiro passo deste teste foi realizado executando o modelo selecionado no primeiro experimento que definia a arquitetura de CNN junto com o otimizador Adamax. O treinamento deste modelo foi parado a partir da época 20, dado que não houve melhorias após esta época considerando os critérios de parada estabelecidos para o treinamento dos modelos em todos os experimentos. Os dados obtidos no treinamento deste modelo são mostrados através da [Figura 52](#).

Figura 52 – Resultados do treinamento usando Adamax



Fonte – Danilo Assunção, 2022

Abaixo é exibido, a partir do Quadro 23, os dados obtidos no teste deste modelo depois da fase de treinamento.

Quadro 23 – Resultados do teste usando Adamax

Épocas	Perda	Acurácia	Precisão	Revocação	Medida F	Acertos	Erros
12	0,08	97,91%	97,91%	97,91%	97,91%	187	4

Fonte – Danilo Assunção, 2022

Considerando os resultados obtidos, foram de fato relevantes, porém se comparado com o modelo utilizando SGD, o Adamax obteve uma Medida F e Acurácia maior, atingindo uma baixa diferença de 1,58% em ambas as medidas, porém com 6 épocas de treinamento a mais se comparado com o modelo que usou SGD.

5.2.4 Conclusão

Neste segundo experimento foi feito testes utilizando alguns algoritmos de otimização junto com a ResNet-152 (arquitetura selecionada no primeiro experimento), visando avaliar

quais destes métodos fornecem o melhor resultado com base no problema que envolve a identificação das espécies de abelhas.

No escopo analisado, os algoritmos Adadelta e Adagrad demonstraram o menor desempenho em relação a todos os algoritmos avaliados. Neste último experimento, dois algoritmos obtiveram valores acima de 96% de Acurácia e Medida F, sendo tais algoritmos, SGD e Adamax. O algoritmo Adamax foi o que atingiu a maior Acurácia e Medida F em comparação ao SGD, com uma diferença de 1,58% para ambas as medidas citadas, ainda assim, isto foi obtido com um custo de 6 ciclos de treinamento adicionais, dado que o SGD foi o que convergiu mais rapidamente. Contudo, foi perceptível a vantagem no uso do Adamax que possui a capacidade de determinar a taxa de aprendizado de maneira automática com base no cenário abordado e este aspecto acaba por ser uma vantagem com relação à algoritmos como o SGD, dado que este tipo de algoritmo exige o ajuste manual de seus hiperparâmetros, como a taxa de aprendizado, o que torna mais complexo determinar o valor ideal neste contexto (LE *et al.*, 2011). A partir disto, foi determinado que o algoritmo selecionado neste experimento é o Adamax, que alcançou uma Acurácia e Medida F de 97,91%.

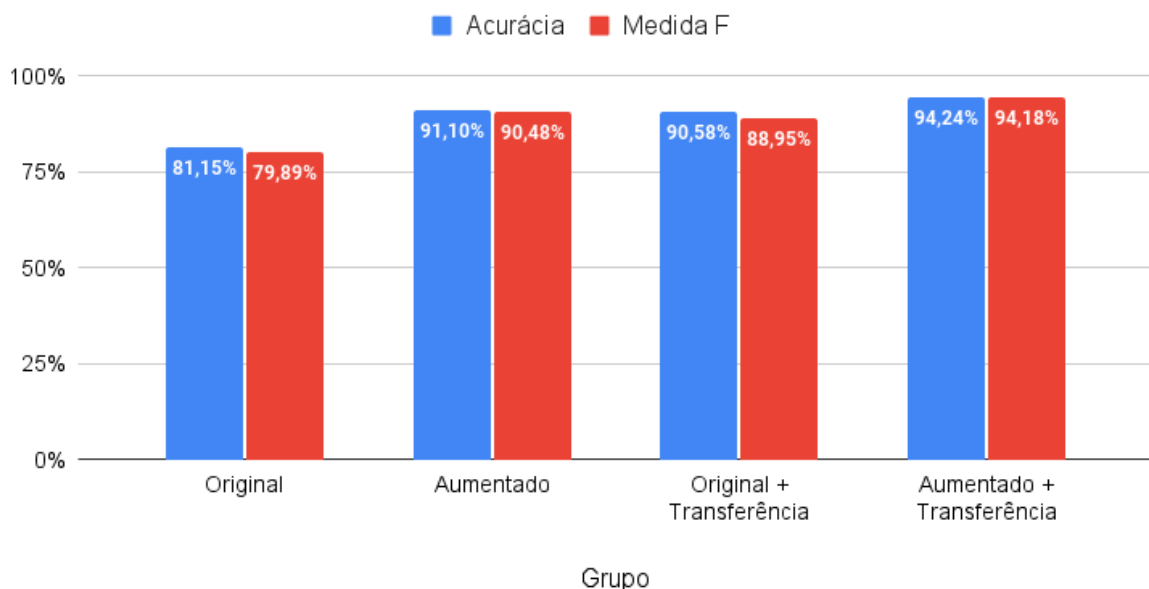
5.3 Modelo final

Para determinar o modelo final, foram conduzidos dois experimentos essenciais: a seleção da arquitetura de uma CNN e a seleção do algoritmo de otimização para a arquitetura selecionada e, com isso, foi possível determinar os componentes para a construção de um modelo em relação ao domínio abordado. No primeiro experimento, a arquitetura selecionada foi a ResNet-152 em conjunto com a técnica de transferência de aprendizado proposta neste trabalho, e esta arquitetura possibilitou o modelo a obter um resultado de 96,33% nas medidas comparativas de Acurácia e Medida F. E para o segundo experimento, o que melhor desempenhou em termos de Acurácia e Medida F, junto à ResNet-152, foi o algoritmo Adamax que obteve respectivamente para ambas as medidas com um total de 97,91%. Portanto, para o modelo final, foi definido utilizando a ResNet-152 com o algoritmo Adamax aplicando a técnica de aprendizado por transferência em cima do conjunto de dados ampliado, através da técnica de aumento de dados.

5.4 Conclusão dos resultados

Este capítulo detalhou todos os cenários analisados que levaram à escolha final do modelo com melhor desempenho. Com base nos valores anteriormente apresentados foi possível observar que, o uso de uma CNN com a utilização do conjunto de dados original, sem empregar nenhuma técnica específica para aprimoramento dos resultados, não seria possível alcançar valores relevantes devido ao seu baixo volume de dados, porém a utilização de técnicas de aumento de dados e transferência de aprendizado possibilitaram o uso de uma CNN para o domínio abordado. Para uma melhor visualização desta percepção, a seguir são apresentados alguns gráficos através das Figuras 53 e 54, que apresentam a média aritmética dos valores obtidos nas análises feitas através do primeiro experimento para determinados grupos de experimentação, sendo tais grupos, o conjunto de dados original (Original), o conjunto de dados aumentado (Aumentado), o conjunto de dados original com transferência de aprendizado (Original + Transferência) e o conjunto de dados aumentado com transferência de aprendizado (Aumentado + Transferência).

Figura 53 – Média aritmética de acurácia e medida F por Grupo



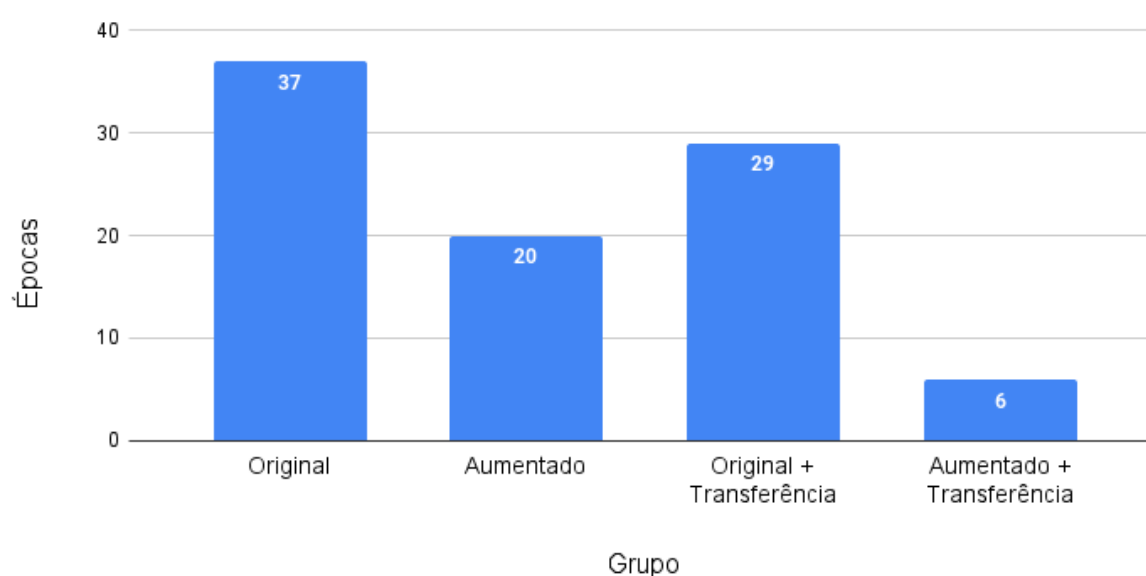
Fonte – Danilo Assunção, 2022

Na Figura 53 temos a visualização das médias aritméticas de Acurácia e Medida F de cada grupo de experimentação, permitindo visualizar a influência das técnicas de aumento de dados e transferência de aprendizado nos experimentos em geral. O uso do

conjunto de dados original, sem a aplicação de quaisquer técnicas, obteve em média 81,15% na Acurácia e 79,89% na Medida F, respectivamente, o mesmo conjunto de dados com o uso da transferência de aprendizado, resultaram na subida, em média, de 9,43% na Acurácia e 9,06% na Medida F. Neste caso, a transferência de aprendizado indicou aprimorar os resultados, em média, um pouco acima de 9% para ambas as medidas e, dado o aumento da Medida F, indicando que este método também melhorou a precisão e revocação dos modelos testados. Para o grupo que usou o conjunto de dados aumentado, foi obtido nos testes em média uma Acurácia e Medida F de 91,10% e 90,48%, respectivamente, neste mesmo conjunto de dados com a união da técnica de transferência de aprendizado, as duas medidas subiram para 94,24% e 94,18%, ocorrendo um aumento em média de 4% na Acurácia e Medida F. Analisando o grupo inicial, sem nenhum uso das técnicas propostas com o último grupo que utilizou ambas as técnicas de aumento de dados e transferência de aprendizado, é alcançado, em média, uma diferença de 13,09% na Acurácia e 14,29% na Medida F, o que enfatiza uma notável melhora quando tais técnicas são empregadas em um contexto com um conjunto de dados pequeno.

Não somente as medidas usadas para avaliar o modelo foram melhoradas, mas também a quantidade média de épocas necessárias para se obter tais resultados, conforme é apresentado pela Figura 54.

Figura 54 – Média de Épocas por Grupo



Neste gráfico que relaciona a quantidade média de épocas usadas no treinamento com relação aos grupos de experimentação deste trabalho, é possível observar que as técnicas de aumento de dados e transferência de aprendizado tiveram um impacto direto na diminuição da quantidade de épocas médias necessárias para se atingir os melhores resultados dos modelos, limitado ao valor de até 50 épocas. Considerando o grupo com o uso médio de épocas maior e menor, obteve-se uma diferença de 31 épocas (-83,78%), demonstrando que as técnicas de aumento de dados e transferência de aprendizado, em conjunto, não somente melhoraram os resultados do modelo como também precisaram de menos épocas para obter estes resultados.

Concluindo, foi observado a partir dos experimentos que a aplicação de uma CNN pode ser uma ferramenta bem poderosa para o problema que envolve o reconhecimento das espécies de abelhas, contudo, dado o cenário com um baixo volume de dados, uma CNN pode não exibir bons resultados e, para solucionar este contexto, foi feito o uso de técnicas de aumento de dados e transferência de aprendizado, que auxiliam no aumento do desempenho do resultado do modelo proposto e permitindo que o modelo final selecionado obtivesse uma Acurácia e Medida F de 97,91% nos testes executados.

6 Conclusão

O problema inicial que aborda a classificação das espécies de abelhas através da imagem de suas asas surgiu baseado em necessidades reais. Atualmente o responsável por realizar a classificação dessas espécies é um especialista que necessita de um vasto conhecimento para conseguir realizar essa tarefa que é totalmente manual, então dois problemas podem ser percebidos com isto, o primeiro é a necessidade de um especialista no assunto, o qual pode haver certa dificuldade de se encontrar e o segundo é o trabalho manual que é suscetível a erros. Diante disto, foi visto que através da imagem das asas de abelhas podem ser desenvolvidos métodos estatísticos que permitem realizar a classificação destas espécies, tais métodos podem ser realizados através de aprendizado de máquina que permitem automatizar a tarefa de identificação das espécies de maneira computadorizada.

Com base nesta premissa, em [Rebelo *et al.* \(2021\)](#) foi desenvolvido um importante trabalho que relaciona métodos que envolvem o pré-processamento e segmentação das imagens para extração de características relevantes contidas nestes componentes e a classificação automática destas espécies utilizando um algoritmo de aprendizado de máquina tradicional. Tal prática resultou em cenários promissores que demonstram resultados bem relevantes. Contudo, ainda mirando na evolução destes métodos, propusemos o uso de algoritmos mais sofisticados incluídos no campo de aprendizado profundo para a classificação das espécies de abelhas, neste caso, algoritmos que permitam tal ação sem mais a necessidade do uso de métodos *handcrafted* no processo que extrai as características das imagens, e o método escolhido para essa evolução foi viabilizar o uso de uma CNN.

Todavia, foi identificado que o uso de uma CNN envolve algumas premissas como uma quantidade grande de dados para se obter bons resultados e também há certa complexidade em sua execução. Para minimizar tais complexidades foi estudado neste trabalho certas técnicas encontradas na literatura que permitem driblar cenários que estão relacionados com o problema que é a baixa quantidade de dados disponíveis para se executar o treinamento adequado do modelo usando uma CNN e essas técnicas denominadas como técnica de aumento de dados e técnica de transferência de aprendizado foram utilizadas nos experimentos realizados e possibilitaram o uso de um método mais avançado em termos de extração de características e classificação atingindo resultados satisfatórios (97,91% de Acurácia e Medida F) relacionados com o da *baseline* definido neste trabalho.

6.1 Limitações

Em maioria, quase todos os hiperparâmetros considerados neste trabalho foram usados de maneira padrão fornecidos pela biblioteca utilizada na ferramenta proposta, ou seja, ainda existe um espaço para melhorias dos parâmetros de otimização. E outra limitação neste trabalho tem relação com a quantidade de imagens disponibilizadas para testes, devido ao conjunto de dados disponibilizado originalmente ter uma quantidade pequena de imagens disponíveis.

6.2 Trabalhos futuros

Abaixo são apresentadas algumas possibilidades que podem ser futuramente aplicadas como evolução deste projeto.

- **Estudo de métodos que auxiliem na identificação de espécies de abelhas sem ferrão:** como citado na introdução, a identificação automática pode ser um desafio, principalmente para as abelhas sem ferrão que possuem características muito parecidas entre as espécies de mesmo grupo ([FRANCOY; IMPERATRIZ-FONSECA, 2010](#)) e isso de fato se mostrou válido dado que neste trabalho os resultados menos precisos foram relacionados ao grupo de espécie de abelhas sem ferrão denominado *Austroplebeia*. É viável que seja feito um estudo para encontrar algum método que permita que uma CNN consiga classificar espécies dentro destes grupos de espécies com maior precisão;
- **GAN para o aumento do conjunto de dados:** em [Shorten e Khoshgoftaar \(2019\)](#) é proposto um método sofisticado utilizando GANs (Redes adversárias generativas). A partir deste método, pode haver a criação de um método que irá permitir ter uma rede especializada na geração de imagens de asas de abelhas que ajudarão no crescimento do conjunto de dados;
- **Hiperparâmetros do otimizador da CNN:** utilização de métodos que permitam identificar os melhores hiperparâmetros para o algoritmo de otimização da CNN visando aumentar o seu desempenho e com isto aumentar conseqüentemente o desempenho da rede;

- **Teste de outros métodos de transferência de aprendizado:** em [Li e Hoiem \(2017\)](#) são definidas algumas técnicas além do método de extração de características e ajuste fino que podem também demonstrar bons resultados se aplicado neste domínio de identificação de espécies;
- **Arquitetura automática da CNN:** dada a complexidade contida em arquiteturas de CNN, alguns trabalhos estão realizando estudos que visam determinar a um domínio de maneira totalmente automática e não mais utilizando estruturas de CNN já pré-estabelecidas como as aplicadas neste trabalho. Um cenário é uso de algoritmos genéticos para a criação de arquiteturas de CNN de maneira automática como apresentado no trabalho de [Ahmed, Darwish e El-Sherbiny \(2019\)](#).

Referências

- AHMED, A. A.; DARWISH, S. M. S.; EL-SHERBINY, M. M. A novel automatic cnn architecture design approach based on genetic algorithm. In: SPRINGER. *International Conference on Advanced Intelligent Systems and Informatics*. [S.l.], 2019. p. 473–482. Citado na página 109.
- ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: IEEE. *2017 International Conference on Engineering and Technology (ICET)*. [S.l.], 2017. p. 1–6. Citado na página 35.
- BANAN, A.; NASIRI, A.; TAHERI-GARAVAND, A. Deep learning-based appearance features extraction for automated carp species identification. *Aquacultural Engineering*, Elsevier, v. 89, p. 102053, 2020. Citado na página 47.
- BONACCORSO, G. *Machine learning algorithms*. [S.l.]: Packt Publishing Ltd, 2017. Citado na página 27.
- BUSCHBACHER, K.; AHRENS, D.; ESPELAND, M.; STEINHAGE, V. Image-based species identification of wild bees using convolutional neural networks. *Ecological Informatics*, Elsevier, v. 55, p. 101017, 2020. Citado na página 71.
- CHEN, Y.; ZHANG, X.; CHEN, W.; LI, Y.; WANG, J. Research on recognition of fly species based on improved retinanet and cbam. *IEEE Access*, IEEE, v. 8, p. 102907–102919, 2020. Citado na página 47.
- CIBUK, M.; BUDAK, U.; GUO, Y.; INCE, M. C.; SENGUR, A. Efficient deep features selections and classification for flower species recognition. *Measurement*, Elsevier, v. 137, p. 7–13, 2019. Citado na página 47.
- DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, v. 12, n. 7, 2011. Citado na página 98.
- FENG, V. An overview of resnet and its variants. *Towards data science*, v. 2, 2017. Citado na página 74.
- FRANCOY, T. M.; IMPERATRIZ-FONSECA, V. L. A morfometria geométrica de asas ea identificação automática de espécies de abelhas. *Oecologia Australis*, v. 14, n. 1, p. 317–321, 2010. Citado 2 vezes nas páginas 19 e 108.
- FRANCOY, T. M.; WITTMANN, D.; DRAUSCHKE, M.; MÜLLER, S.; STEINHAGE, V.; BEZERRA-LAURE, M. A.; JONG, D. D.; GONÇALVES, L. S. Identification of africanized honey bees through wing morphometrics: two fast and efficient procedures. *Apidologie*, EDP Sciences, v. 39, n. 5, p. 488–494, 2008. Citado na página 19.
- GÓMEZ-RÍOS, A.; TABIK, S.; LUENGO, J.; SHIHAVUDDIN, A.; HERRERA, F. Coral species identification with texture or structure images using a two-level classifier based on convolutional neural networks. *Knowledge-Based Systems*, Elsevier, v. 184, p. 104891, 2019. Citado 2 vezes nas páginas 47 e 54.

- GONZALEZ, R. C. Deep convolutional neural networks [lecture notes]. *IEEE Signal Processing Magazine*, IEEE, v. 35, n. 6, p. 79–87, 2018. Citado 2 vezes nas páginas 20 e 23.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. [S.l.]: Editora Blucher, 2000. Citado na página 23.
- GYIRES-TÓTH, B. P.; OSVÁTH, M.; PAPP, D.; SZÚCS, G. Deep learning for plant classification and content-based image retrieval. *Cybernetics and Information Technologies*, Sciendo, v. 19, n. 1, p. 88–100, 2019. Citado na página 48.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado na página 83.
- HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, World Scientific, v. 6, n. 02, p. 107–116, 1998. Citado na página 74.
- HOLGADO, A. G.; PABLOS, S. M.; PEÑALVO, F. J. G. *et al.* Guidelines for performing systematic research projects reviews. *International Journal of Interactive Multimedia and Artificial Intelligence*, v. 6, n. 2, p. 9, 2020. Citado na página 41.
- HONG, S.; MEHDI, S. R.; HUANG, H.; SHAHANI, K.; ZHANG, Y.; JUNAIDULLAH, K. R.; KHAN, M. A. Classification of freshwater zooplankton by pre-trained convolutional neural network in underwater microscopy. 2020. Citado 2 vezes nas páginas 47 e 57.
- HSIANG, A. Y.; BROMBACHER, A.; RILLO, M. C.; MLENECK-VAUTRAVERS, M. J.; CONN, S.; LORDSMITH, S.; JENTZEN, A.; HENEHAN, M. J.; METCALFE, B.; FENTON, I. S. *et al.* Endless forams: 34,000 modern planktonic foraminiferal images for taxonomic training and automated species recognition using convolutional neural networks. *Paleoceanography and Paleoclimatology*, Wiley Online Library, v. 34, n. 7, p. 1157–1177, 2019. Citado na página 48.
- KHALIFA, N. E. M.; TAHA, M. H. N.; HASSANIEN, A. E.; HEMEDAN, A. A. Deep bacteria: robust deep learning data augmentation design for limited bacterial colony dataset. *International Journal of Reasoning-based Intelligent Systems*, Inderscience Publishers (IEL), v. 11, n. 3, p. 256–264, 2019. Citado na página 48.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 101.
- LAUZON, F. Q. An introduction to deep learning. In: *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*. [S.l.: s.n.], 2012. p. 1438–1439. Citado na página 29.
- LE, Q. V.; NGIAM, J.; COATES, A.; LAHIRI, A.; PROCHNOW, B.; NG, A. Y. On optimization methods for deep learning. In: *ICML*. [S.l.: s.n.], 2011. Citado na página 103.
- LE, V.-L.; BEURTON-AIMAR, M.; ZEMMARI, A.; MARIE, A.; PARISEY, N. Automated landmarking for insects morphometric analysis using deep neural networks. *Ecological Informatics*, Elsevier, v. 60, p. 101175, 2020. Citado 2 vezes nas páginas 20 e 46.

- LEE, S. H.; CHAN, C. S.; MAYO, S. J.; REMAGNINO, P. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition*, Elsevier, v. 71, p. 1–13, 2017. Citado na página [49](#).
- LI, K.; LIN, J.; LIU, J.; ZHAO, Y. Using deep learning for image-based different degrees of ginkgo leaf disease classification. *Information*, Multidisciplinary Digital Publishing Institute, v. 11, n. 2, p. 95, 2020. Citado 2 vezes nas páginas [47](#) e [56](#).
- LI, R.; WANG, R.; ZHANG, J.; XIE, C.; LIU, L.; WANG, F.; CHEN, H.; CHEN, T.; HU, H.; JIA, X. *et al.* An effective data augmentation strategy for cnn-based pest localization and recognition in the field. *IEEE Access*, IEEE, v. 7, p. 160274–160283, 2019. Citado na página [47](#).
- LI, Z.; HOIEM, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 40, n. 12, p. 2935–2947, 2017. Citado 4 vezes nas páginas [38](#), [54](#), [68](#) e [109](#).
- LIM, S.; KIM, S.; KIM, D. Performance effect analysis for insect classification using convolutional neural network. In: IEEE. *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. [S.l.], 2017. p. 210–215. Citado 2 vezes nas páginas [41](#) e [49](#).
- LIU, B.; DING, Z.; TIAN, L.; HE, D.; LI, S.; WANG, H. Grape leaf disease identification using improved deep convolutional neural networks. *Frontiers in Plant Science*, Frontiers, v. 11, p. 1082, 2020. Citado na página [46](#).
- LIU, S.; SHIH, F. Y.; RUSSELL, G.; RUSSELL, K.; PHAN, H. Classification of ecological data by deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific, v. 34, n. 13, p. 2052010, 2020. Citado 5 vezes nas páginas [20](#), [41](#), [46](#), [51](#) e [62](#).
- LU, Y.-C.; TUNG, C.; KUO, Y.-F. Identifying the species of harvested tuna and billfish using deep convolutional neural networks. *ICES Journal of Marine Science*, Oxford University Press, v. 77, n. 4, p. 1318–1329, 2020. Citado na página [46](#).
- MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of machine learning*. [S.l.]: MIT press, 2018. Citado 2 vezes nas páginas [24](#) e [25](#).
- NASTESKI, V. An overview of the supervised machine learning methods. *Horizons. b*, v. 4, p. 51–62, 2017. Citado na página [26](#).
- O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015. Citado 6 vezes nas páginas [26](#), [27](#), [30](#), [32](#), [33](#) e [36](#).
- PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, IEEE, v. 22, n. 10, p. 1345–1359, 2009. Citado na página [37](#).
- POTTS, S. G.; IMPERATRIZ-FONSECA, V.; NGO, H. T.; BIESMEIJER, J. C.; BREEZE, T. D.; DICKS, L. V.; GARIBALDI, L. A.; HILL, R.; SETTELE, J.; VANBERGEN, A. J. *The assessment report on pollinators, pollination and food production: summary for policymakers*. [S.l.]: Secretariat of the Intergovernmental Science-Policy Platform on Biodiversity . . . , 2016. Citado na página [19](#).

- RAUF, H. T.; LALI, M. I. U.; ZAHOOR, S.; SHAH, S. Z. H.; REHMAN, A. U.; BUKHARI, S. A. C. Visual features based automated identification of fish species using deep convolutional neural networks. *Computers and Electronics in Agriculture*, Elsevier, v. 167, p. 105075, 2019. Citado na página 47.
- REBELO, A. R.; FAGUNDES, J. M.; DIGIAMPIETRI, L. A.; FRANCOY, T. M.; BÍSCARO, H. H. A fully automatic classification of bee species from wing images. *Apidologie*, Springer, v. 52, n. 6, p. 1060–1074, 2021. Citado 4 vezes nas páginas 19, 21, 23 e 107.
- REN, F.; LIU, W.; WU, G. Feature reuse residual networks for insect pest recognition. *IEEE Access*, IEEE, v. 7, p. 122758–122768, 2019. Citado na página 48.
- ROCKSTRÖM, J.; STEFFEN, W.; NOONE, K.; PERSSON, Å.; CHAPIN, F. S.; LAMBIN, E. F.; LENTON, T. M.; SCHEFFER, M.; FOLKE, C.; SCHELLNHUBER, H. J. *et al.* A safe operating space for humanity. *nature*, Nature Publishing Group, v. 461, n. 7263, p. 472–475, 2009. Citado na página 19.
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATY, A.; KHOSLA, A.; BERNSTEIN, M. *et al.* Imagenet large scale visual recognition challenge. *International journal of computer vision*, Springer, v. 115, n. 3, p. 211–252, 2015. Citado na página 37.
- SHAO, L.; ZHU, F.; LI, X. Transfer learning for visual categorization: A survey. *IEEE transactions on neural networks and learning systems*, IEEE, v. 26, n. 5, p. 1019–1034, 2014. Citado na página 38.
- SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, Springer, v. 6, n. 1, p. 1–48, 2019. Citado 3 vezes nas páginas 38, 65 e 108.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado na página 88.
- SUN, Y.; LIU, Y.; WANG, G.; ZHANG, H. Deep learning for plant identification in natural environment. *Computational intelligence and neuroscience*, Hindawi, v. 2017, 2017. Citado 2 vezes nas páginas 49 e 57.
- SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Thirty-first AAAI conference on artificial intelligence*. [S.l.: s.n.], 2017. Citado na página 73.
- SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2818–2826. Citado na página 78.
- TAN, J. wei; CHANG, S.-W.; ABDUL-KAREEM, S.; YAP, H. J.; YONG, K.-T. Deep learning for plant species classification using leaf vein morphometric. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, IEEE, v. 17, n. 1, p. 82–90, 2018. Citado na página 48.

TETILA, E. C.; MACHADO, B. B.; MENEZES, G. V.; BELETE, N. A. de S.; ASTOLFI, G.; PISTORI, H. A deep-learning approach for automatic counting of soybean insect pests. *IEEE Geoscience and Remote Sensing Letters*, IEEE, v. 17, n. 10, p. 1837–1841, 2019. Citado na página 48.

THENMOZHI, K.; REDDY, U. S. Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture*, Elsevier, v. 164, p. 104906, 2019. Citado 2 vezes nas páginas 48 e 52.

TIWARI, S. A comparative study of deep learning models with handcraft features and non-handcraft features for automatic plant species identification. *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, IGI Global, v. 11, n. 2, p. 44–57, 2020. Citado na página 47.

UHRIG, R. Introduction to artificial neural networks. In: *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics*. [S.l.: s.n.], 1995. v. 1, p. 33–37 vol.1. Citado na página 30.

VALAN, M.; MAKONYI, K.; MAKI, A.; VONDRÁČEK, D.; RONQUIST, F. Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks. *Systematic Biology*, Oxford University Press, v. 68, n. 6, p. 876–895, 2019. Citado na página 48.

VILLON, S.; MOUILLOT, D.; CHAUMONT, M.; DARLING, E. S.; SUBSOL, G.; CLAVERIE, T.; VILLÉGER, S. A deep learning method for accurate and fast identification of coral reef fishes in underwater images. *Ecological informatics*, Elsevier, v. 48, p. 238–244, 2018. Citado na página 49.

WANG, R.; ZHANG, J.; DONG, W.; YU, J.; XIE, C. J.; LI, R.; CHEN, T.; CHEN, H. A crop pests image classification algorithm based on deep convolutional neural network. *Telkomnika*, v. 15, n. 3, p. 1239–1246, 2017. Citado na página 49.

YAMASHITA, R.; NISHIO, M.; DO, R. K. G.; TOGASHI, K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, Springer, v. 9, n. 4, p. 611–629, 2018. Citado 5 vezes nas páginas 31, 32, 34, 36 e 37.

YU, D.; WANG, H.; CHEN, P.; WEI, Z. Mixed pooling for convolutional neural networks. In: SPRINGER. *International conference on rough sets and knowledge technology*. [S.l.], 2014. p. 364–375. Citado na página 36.

ZAHEER, R.; SHAZIYA, H. A study of the optimization algorithms in deep learning. In: IEEE. *2019 third international conference on inventive systems and control (ICISC)*. [S.l.], 2019. p. 536–539. Citado na página 30.

ZEILER, M. D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. Citado na página 100.

ZHANG, X.; CHEN, G. An automatic insect recognition algorithm in complex background based on convolution neural network an automatic insect recognition algorithm in complex background based on convolution neural network. Citado na página 46.

ZHU, H.; LIU, Q.; QI, Y.; HUANG, X.; JIANG, F.; ZHANG, S. Plant identification based on very deep convolutional neural networks. *Multimedia Tools and Applications*, Springer, v. 77, n. 22, p. 29779–29797, 2018. Citado na página 49.

ZHU, X.; ZHU, M.; REN, H. Method of plant leaf recognition based on improved deep convolutional neural network. *Cognitive Systems Research*, Elsevier, v. 52, p. 223–233, 2018. Citado na página [49](#).

ZIELIŃSKI, B.; PLICHTA, A.; MISZTAL, K.; SPUREK, P.; BRZYCHCZY-WŁOCH, M.; OCHOŃSKA, D. Deep learning approach to bacterial colony classification. *PloS one*, Public Library of Science San Francisco, CA USA, v. 12, n. 9, p. e0184554, 2017. Citado na página [49](#).

Apêndice B – Medidas obtidas no modelo final por espécie

Espécie	VP	FP	FN	VN	Acurácia	Precisão	Revocação	Medida F
<i>Apis cerana</i>	4	0	0	187	100%	100%	100%	100%
<i>Apis dorsata</i>	4	0	0	187	100%	100%	100%	100%
<i>Apis mellifera</i>	5	0	0	186	100%	100%	100%	100%
<i>Apis florea</i>	4	0	0	187	100%	100%	100%	100%
<i>Austroplebeia australis</i>	4	0	1	186	99,48%	100%	80%	88,89%
<i>Austroplebeia cincta</i>	4	0	0	187	100%	100%	100%	100%
<i>Austroplebeia essinatoni</i>	2	0	2	187	98,95%	100%	50%	66,67%
<i>Austroplebeia striped</i>	4	2	0	185	98,95%	66,67%	100%	80%
<i>Austroplebeia symei</i>	4	1	0	186	99,48%	80%	100%	88,89%
<i>Axestotrigona ferruginea</i>	4	1	0	186	99,48%	80%	100%	88,89%
<i>Bombus brasiliensis</i>	4	0	0	187	100%	100%	100%	100%
<i>Bombus pauloensis</i>	3	0	0	188	100%	100%	100%	100%
<i>Dactylurina staudingeri</i>	4	0	0	187	100%	100%	100%	100%
<i>Eufriesea violacea</i>	4	0	0	187	100%	100%	100%	100%
<i>Euglossa annectans</i>	4	0	0	187	100%	100%	100%	100%
<i>Euglossa mixta panama</i>	4	0	0	187	100%	100%	100%	100%
<i>Euglossa truncata</i>	3	0	0	188	100%	100%	100%	100%
<i>Eulaema nigrita</i>	2	0	0	189	100%	100%	100%	100%
<i>Exaerete smaragdina</i>	3	0	0	188	100%	100%	100%	100%
<i>Geotrigona sp</i>	4	0	0	187	100%	100%	100%	100%
<i>Lestrimelitta limao</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona bicolor</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona flavolineata</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona mandacaia</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona paratrígona subnuda</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona plebeia nigriceps</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona quadrifasciata</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona scutellaris</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona seminigra</i>	4	0	0	187	100%	100%	100%	100%
<i>Melipona subnitida</i>	4	0	0	187	100%	100%	100%	100%
<i>Meliponula bocandei</i>	4	0	1	186	99,48%	100%	80,00%	88,89%
<i>Mourella caerulea</i>	3	0	0	188	100%	100%	100%	100%
<i>Nannotrigona testaceicornis</i>	4	0	0	187	100%	100%	100%	100%
<i>Partamona helleri</i>	4	0	0	187	100%	100%	100%	100%
<i>Plebeia sp</i>	4	0	0	187	100%	100%	100%	100%
<i>Plebeia droryana</i>	4	0	0	187	100%	100%	100%	100%
<i>Plebeia flavocincta</i>	5	0	0	186	100%	100%	100%	100%
<i>Plebeia pugnax</i>	4	0	0	187	100%	100%	100%	100%
<i>Plebeia remota</i>	4	0	0	187	100%	100%	100%	100%
<i>Scaptotrigona bipunctata</i>	4	0	0	187	100%	100%	100%	100%
<i>Scaptotrigona depilis</i>	4	0	0	187	100%	100%	100%	100%
<i>Scaptotrigona tubiba</i>	4	0	0	187	100%	100%	100%	100%
<i>Scaura latitarsis</i>	4	0	0	187	100%	100%	100%	100%
<i>Schwarziana quadripunctata</i>	5	0	0	186	100%	100%	100%	100%
<i>Trigona spinipes</i>	4	0	0	187	100%	100%	100%	100%
<i>Tripoxylon aurifrons</i>	4	0	0	187	100%	100%	100%	100%
<i>Tripoxylon lactitarse</i>	4	0	0	187	100%	100%	100%	100%
<i>Tripoxylon rogenhoferi</i>	4	0	0	187	100%	100%	100%	100%

Apêndice C – Medidas obtidas no modelo final por gênero de espécies

Gênero	VP	FP	FN	VN	Acurácia	Precisão	Revocação	Medida F
Apis	17	0	0	747	100%	100%	100%	100%
Austroplebeia	18	3	3	931	99,37%	85,71%	85,71%	85,71%
Axestotrigona	4	1	0	186	99,48%	80%	100%	88,89%
Bombus	7	0	0	375	100%	100%	100%	100%
Dactylurina	4	0	0	187	100%	100%	100%	100%
Eufriesea	4	0	0	187	100%	100%	100%	100%
Euglossa	11	0	0	562	100%	100%	100%	100%
Eulaema	2	0	0	189	100%	100%	100%	100%
Exaerete	3	0	0	188	100%	100%	100%	100%
Feotrigona	4	0	0	187	100%	100%	100%	100%
Lestrimellita	4	0	0	187	100%	100%	100%	100%
Plebeia	21	0	0	934	100%	100%	100%	100%
Melipona	36	0	0	1.683	100%	100%	100%	100%
Meliponula	4	0	1	186	99,48%	100%	80%	88,89%
Mourella	3	0	0	188	100%	100%	100%	100%
Nannotrigona	4	0	0	187	100%	100%	100%	100%
Partamona	4	0	0	187	100%	100%	100%	100%
Scaptotrigona	12	0	0	561	100%	100%	100%	100%
Scaura	4	0	0	187	100%	100%	100%	100%
Schwarziana	5	0	0	186	100%	100%	100%	100%
Trigona	4	0	0	187	100%	100%	100%	100%
Tripoxyton	12	0	0	561	100%	100%	100%	100%