



**UM MODELO DE APRENDIZAGEM PROFUNDA  
PARA RECONHECIMENTO DE PRAGAS EM  
MILHARAS**

**Witenberg Santiago Rodrigues Souza**

**Dissertação de Mestrado  
Sistemas Mecatrônicos**

**UNIVERSIDADE DE BRASÍLIA**

**Faculdade de Tecnologia  
Departamento de Engenharia Mecânica**

UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

UM MODELO DE APRENDIZAGEM PROFUNDA PARA  
RECONHECIMENTO DE PRAGAS EM MILHARAI

Witenberg Santiago Rodrigues Souza

Orientador: Díbio Leandro Borges, Dr. (CIC/ UnB)

DISSERTAÇÃO DE MESTRADO

PUBLICAÇÃO: ENM.DM - XXX.XXX

BRASÍLIA/DF: 30 de julho de 2019

UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Um Modelo de Aprendizagem Profunda para Reconhecimento  
de Pragas em Milharais

Witenberg Santiago Rodrigues Souza

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE  
ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNI-  
VERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS PARCIAIS  
PARA A OBTENÇÃO DO GRAU DE MESTRE EM SISTEMAS MECATRÔ-  
NICOS.

APROVADA POR:

---

Díbio Leandro Borges, Dr. (CIC/ UnB)  
(Orientador)

---

José Maurício Torres da Motta, Dr. (ENM/ UnB)  
(Examinador Interno)

---

Geraldo Pereira Rocha Filho, Dr. (CIC/ UnB)  
(Examinador Externo)

BRASÍLIA/DF, 30 DE JULHO DE 2019.

## FICHA CATALOGRÁFICA

Souza, W. S. R.

Um Modelo de Aprendizagem Profunda para Reconhecimento de Pragas em Milharais [Distrito Federal] 2019.

xvii, 84p. (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2019.

Dissertação de Mestrado - Universidade de Brasília.

Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

Palavras-chave:

1. Reconhecimento de Insetos

2. Visão por Computador

3. Aprendizagem Residual Profunda

4. Pragas do Milho

I. ENM/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

Souza, W. S. R.(2019). Um Modelo de Aprendizagem Profunda para Reconhecimento de Pragas em Milharais. Dissertação de Mestrado, Publicação ENM.DM - XXX.XXX, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, Distrito Federal, xvii, 84p.

## CESSÃO DE DIREITOS

**NOME DO AUTOR:** Witenberg Santiago Rodrigues Souza.

**TÍTULO DA DISSERTAÇÃO DE MESTRADO:** Um Modelo de Aprendizagem Profunda para Reconhecimento de Pragas em Milharais.

**GRAU / ANO:** MESTRE / 2019

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.

---

Witenberg Santiago Rodrigues Souza

*Dedico a minha mãe Nilza C. Rodrigues, a minha avó Margarida C., e a Vanessa S. Rodrigues, minha irmã. Sem elas não, nunca chegaria a ser o primeiro formando universitário de minha casa. E apresento de volta a Deus o resultado da oportunidade orquestrada por Ele.*

# Agradecimentos

O louvor dado a alguém por seus feitos, não-raro pode ser estendido aos que o acompanharam desde o início até a consumação de sua conquista. Por isso, expresso aqui meus sinceros agradecimentos a Deus pelo dom da vida e por sua orientação diária e conforto nos momentos difíceis. Agradeço a minha família, Nilza, minha mãe, batalhadora; minha avó, Margarida, grande exemplo de vida; minha irmã, Vanessa, exemplo de garra.

Ao meu orientador, *Professor Díbio Leandro Borges*, agradeço a confiança e por acreditar em meu potencial, quando eu mesmo estava em baixa. Agradeço por compartilhar sua sabedoria e seu conhecimento bem como por todo o incentivo.

A amigos, responsáveis por valioso apoio neste empreendimento, meus sinceros agradecimentos nominalmente a: Professora Onília Cristina, Professor Cícero Almeida, Amanda Letícia, Adão Nunes e ao Grupo Deep Learning Brasília do Instituto Serzedello, TCU, pela oportunidade de poder compartilhar as experiências em *Deep Learning*. Igualmente, agradeço à CAPES por seu apoio durante o mestrado.

# Resumo

**Um Modelo de Aprendizagem Profunda para Reconhecimento de Pragas em Milharais**

**Autor: Witenberg Santiago Rodrigues Souza**

**Orientador: DÍbio Leandro Borges, Dr. (CIC/ UnB)**

**Programa de Pós-graduação em Sistemas Mecatrônicos**

**Brasília, 30 de julho de 2019**

Este trabalho aborda o reconhecimento de pestes artrópodes em milharais por imagens de campo. Primeiramente, construiu-se um novo conjunto de imagens de pragas primárias e secundárias, tomadas em campo aberto. O banco contém imagens originais e ampliadas em número, as quais foram empregadas em uma abordagem de classificação supervisionada de imagens. Foram propostas também, modificações para um modelo original residual de aprendizagem profunda (Inception-V3), doravante chamado Inception-V3\*. As modificações realizadas no modelo possibilitaram uma maior velocidade de aprendizagem, i.e., menor quantidade de épocas de treinamento para alcançar o menor erro quadrático, além da maior exatidão em comparação ao modelo original. Os testes foram sub-agrupados em dois experimentos, um com as pragas primárias somente e o segundo com todas as pragas (primárias e secundárias). A eficácia dos modelos convolucionais foi avaliada e os modelos foram comparados a um classificador de padrão linear binário associado a máquinas de vetores de suporte. Em média, o modelo Inception-V3\* proposto obteve a melhor taxa de acerto com 97,0% de precisão usando validação cruzada.

**Palavras-chaves:** Reconhecimento de Insetos; Visão por Computador; Aprendizagem Residual Profunda; Pragas do Milho.

# Abstract

**A Deep Learning Model for Recognition of Pest Insects in Maize Plantations**

**Author: Witenberg Santiago Rodrigues Souza**

**Supervisor: Díbio Leandro Borges, Dr. (CIC/ UnB)**

**Master in Mechatronic Systems**

**Brasília, 2019**

This work approaches the recognition of arthropod pests in maize plantations by field images. Firstly, it was built a new image dataset of primary and secondary pests in open field. The dataset contains original and augmented images, which were used in a supervised classification approach. There were also proposed modifications for a deep learning residual model (Inception-V3), named from here on as Inception-V3\*. The modifications made in the model provided a faster learning rate, i.e., less training epochs to reach the lowest squared error, besides greater precision in comparison to the original model. The tests were clustered in two experiments, one with primary pests only, and the second with all pests (primary and secondary). The efficacy of the CNN models were evaluated and compared to a Linear Binary Pattern followed by SVM classifier. On average the proposed Inception-V3\* model achieved the best accuracy rate of 97.0% using cross validation.

**Key-words:** Insect Recognition; Computer Vision; Deep Residual Learning; Maize Pests.



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	Motivação	1
1.2	Problemática	2
1.3	Objetivos	4
1.4	Metodologia	4
1.5	Organização do Trabalho	6
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>7</b>
2.1	<b>Pragas Primárias e Secundárias na Cultura do Milho</b>	<b>7</b>
2.1.1	Pragas Primárias	8
2.1.1.1	<i>Spodoptera frugiperda</i>	9
2.1.1.2	<i>Agrotis ipsilon</i>	9
2.1.1.3	<i>Diatraea saccharalis</i>	10
2.1.1.4	<i>Helicoverpa zea</i>	10
2.1.2	Pragas Secundárias	11
2.1.2.1	<i>Dalbulus maidis</i>	11
2.1.2.2	<i>Diabrotica speciosa</i>	11
2.1.2.3	Outras pragas secundárias: Percevejos	12
2.2	<b>Estado da Arte em Reconhecimento de Pragas por Imagem</b>	<b>12</b>
2.3	<b>Redes Neurais Artificiais</b>	<b>15</b>
2.3.1	Redes Neurais Clássicas em Aprendizagem de Máquina	16
2.3.1.1	Estrutura fundamental de uma rede neuronal artificial	16
2.3.2	Redes Neurais Convolucionais	17
2.3.2.1	Camada de Convolução	18
2.3.2.2	Camada de combinação - ( <i>pooling</i> )	20
2.3.2.3	Camada Totalmente Conectada	22
2.3.3	Redes Residuais	22
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>26</b>

<b>3.1</b>	<b>Construção de um Conjunto de Imagens de Pragas de milho em campo aberto</b>	<b>26</b>
3.1.1	Aquisição de imagens de pragas	26
3.1.2	Pré-processamento para ampliação do conjunto de Imagens	27
<b>3.2</b>	<b>Arquiteturas de Redes Neurais</b>	<b>28</b>
3.2.1	Modelo de Rede Neuronal Clássica - Máquina de Vetores de Suporte SVM-LBP	28
3.2.2	Modelo de Rede Convolutacional Típica - AlexNet	32
3.2.3	Modelos de Redes Convolucionais Residuais	33
3.2.3.1	Inception-V3	33
3.2.3.2	Proposição a partir de um Modelo Residual Inception-V3*	37
<b>4</b>	<b>EXPERIMENTOS</b>	<b>40</b>
4.1	Uso do conjunto de imagens durante o experimento	40
4.2	Configurações e Recursos dos Modelos de Aprendizagem	41
<b>5</b>	<b>ANÁLISE DE RESULTADOS</b>	<b>47</b>
5.1	Desempenho dos modelos de Aprendizagem sobre o Conjunto de Imagens	47
5.1.1	Resultados em Pragas Primárias	47
5.1.1.1	Resultados individuais por categoria testada em Primárias- Matrizes de Confusão	49
5.1.2	Resultados em todas as Pragas	51
5.1.2.1	Resultados individuais por categoria testada em Todas - Matrizes de Confusão	56
5.1.3	Comparativo de Desempenho da rede modificada no conjunto Original	60
<b>6</b>	<b>CONCLUSÃO</b>	<b>63</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>65</b>
	<b>APÊNDICES</b>	<b>68</b>
	<b>APÊNDICE A – AMOSTRAS DE IMAGENS DA BASE DE CLASSES DE INSETOS</b>	<b>69</b>
A.1	Imagens de Treinamento: Pragas Primárias	69
A.2	Imagens de Treinamento: Secundárias	69

# Lista de Figuras

Figura 1.1 – Principais produções agrícolas para a safra de 2018/19 o relatório do Departamento de Agricultura dos Estados Unidos (USDA)(USDA, 2019b), com exceção ao algodão por não por ser mensurado em milhões de libras. . . . .	2
Figura 1.2 – Distribuição mundial da produção de milho, safra 2019 segundo o relatório do USDA(USDA, 2019a). . . . .	3
Figura 1.3 – Amostras de pragas do milho nas circunstâncias encontradas durante a aquisição das imagens em campo aberto, I) Pragas Primárias: a) <i>Spodoptera frugiperda</i> , b) <i>Helicoverpa zea</i> , c) <i>Diatraea saccharalis</i> , d) <i>Agrotis ipsilon</i> ; II) Secundárias e) <i>Diabrotica speciosa</i> , f) <i>Dalbulus maidis</i> , g) Imagens de milharais sem pragas. Todas as imagens de treinamento utilizadas neste trabalho encontram-se disponíveis no Apêndice A. . . . .	5
Figura 1.4 – Fluxograma das três grandes fases do trabalho. . . . .	5
Figura 2.1 – Principais pragas do milho em seus diferentes estágios. . . . .	7
Figura 2.2 – Amostras de <i>Spodoptera frugiperda</i> encontrada em diferentes partes da planta. cada imagem foi recortada em formato quadrado de $224 \times 224$ . . . . .	9
Figura 2.3 – Amostras de <i>Agrotis ipsilon</i> em localizações no solo próximas à planta. . . . .	10
Figura 2.4 – Amostras de <i>Diatraea saccharalis</i> em localizações frequentes e os danos que estas podem impor à planta. . . . .	10
Figura 2.5 – Amostras de <i>Helicoverpa zea</i> nas pontas das espigas, principal região de ataque praga nas plantas. . . . .	11
Figura 2.6 – Amostras de <i>Dalbulus maidis</i> em localizações no solo próximas à planta. . . . .	11
Figura 2.7 – Amostras de <i>Diabrotica speciosa</i> em localizações no solo próximas à planta. . . . .	12
Figura 2.8 – Estrutura fundamental em uma rede neuronal artificial. . . . .	17
Figura 2.9 – Representação de uma matriz RGB cuja profundidade representa os três canais de um imagem colorida. . . . .	18
Figura 2.10 – Imagem ilustrativa de uma Matriz de imagem multiplicando kernel ou matriz de filtro . . . . .	19

Figura 2.11–Matriz de saída $3 \times 3$ , resultante da convolução entre a matriz de imagem e do filtro ilustrado na figura 2.10. . . . .	20
Figura 2.12–Operação de <i>max pooling</i> entre uma matriz de imagem e um filtro $2 \times 2$ com <i>stride 2</i> . . . . .	21
Figura 2.13–Modelo de uma rede neuronal convolucional típica. . . . .	23
Figura 2.14–Redes convolucionais, a) Típica, b) Residual. . . . .	24
Figura 3.1 – Processos de transformações realizadas em imagens do conjunto de dados, para obter o balanceamento entre as classes. (a) Imagem original, (b) Rotação de $90^\circ$ , (c) Rotação de $-90^\circ$ , (d) Rotação de $180^\circ$ , (e) Zoom aleatório de até 1,5x, (f) Alongamento aleatório, (g) Rotação aleatória máxima dentro de um intervalo de $10^\circ$ e (h) Variação de intensidade de pixel para simular diferentes condições de iluminação. . . . .	29
Figura 3.2 – Principais passos para realização de aprendizagem supervisionada. . . . .	29
Figura 3.3 – Estrutura básica de uma SVM. Uma camada de entrada com elementos potencialmente separáveis em um hiperplano; um conjunto de parâmetros e uma função de hipótese. . . . .	31
Figura 3.4 – Extração de características de padrão binário (LBP) em um canal utilizada com extração de características no modelo SVM. . . . .	31
Figura 3.5 – Modelo AlexNet adaptado com 7 saídas probabilísticas através da função <i>softmax</i> (4 pragas primárias, 2 secundárias, 1 sem pragas.) . . . . .	34
Figura 3.6 – Modelo Inception-V3 adaptado de (SZEGEDY et al., 2016) e compactado para melhor compreensão de sua macroestrutura. As dois marcadores (Aux* e Out*) remetem às modificações propostas e apresentadas em 3.2.3.2, concernentes aos dois classificadores nas saídas do modelo Inception-V3*. A nova saída auxiliar (Aux*) e a principal (Out*) tiveram suas camadas Lineares substituídas por mini redes neuronais. . . . .	36
Figura 3.7 – Novas saídas (Aux* e Out*) do modelo proposto Inception-V3* com camadas de regularização ( <i>batch normalization</i> e <i>Dropout</i> ), redução de parâmetro e ativações (Linear) camadas de ativação( <i>ReLU</i> ). . . . .	39
Figura 4.1 – Transformações alternativas de entrada: a) tamanho e cor b) cor. . . . .	42
Figura 4.2 – Taxa inicial obtida através de varredura dentro do intervalo de $10^{-22}$ a $10^1$ , a) Erro de treinamento onde valor mínimo aponta o final do intervalo de descida de gradiente em ( $10^{-1}$ ) e o valor em uma magnitude abaixo de tal valor indica a taxa inicial, b) gráfico erro de validação para verificar se a taxa inicial escolhida atende também ao conjunto de validação, c) potencial de precisão atual para cada taxa no intervalo testado. . . . .	45
Figura 5.1 – Redução de erro e crescimento da precisão observados no primeiro experimento (Primárias). Para o modelo original Inception-V3, a) e c) respectivamente; e para o modelo proposto Inception-V3* b) e d), respectivamente. . . . .	48

Figura 5.2 – Redução de erro e crescimento da precisão observados no primeiro experimento (Primárias) para o modelo AlexNet. . . . .	49
Figura 5.3 – Desempenho individual no conjunto de pragas primárias para o modelo SVM-LBP. . . . .	50
Figura 5.4 – Desempenho individual no conjunto de pragas primárias para o modelo AlexNet. . . . .	51
Figura 5.5 – Desempenho individual no conjunto de pragas primárias para o modelo Inception-V3. . . . .	52
Figura 5.6 – Desempenho individual no conjunto de pragas primárias para o modelo Inception-V3*. . . . .	53
Figura 5.7 – Redução de erro e crescimento da precisão observados no primeiro experimento (Todas) para o modelo AlexNet. . . . .	54
Figura 5.8 – Redução de erro e crescimento da precisão observados no primeiro experimento (Todas). Para o modelo original Inception-V3, a) e c) respectivamente; e para o modelo proposto Inception-V3* b) e d), respectivamente. . . . .	55
Figura 5.9 – Desempenho individual no conjunto de pragas (Todas) para o modelo SVM-LBP. . . . .	56
Figura 5.10–Desempenho individual no conjunto de pragas (Todas) para o modelo AlexNet. . . . .	57
Figura 5.11–Desempenho individual no conjunto de pragas (Todas) para o modelo Inception-V3. . . . .	58
Figura 5.12–Desempenho individual no conjunto de pragas todas para o modelo Inception-V3*. . . . .	59
Figura 5.13–Comportamento de cada modelo expresso pelos valores de <i>F-score</i> a) Pragas primárias b) Todas as pragas. . . . .	61
Figura 5.14–Resultados Comparativos de erro e precisão da rede Inception-3* nos conjuntos originais e ampliado: a) erros de treinamento e de validação (originais), b) precisão para o conjunto original de imagens, c) erros de treinamento e de validação (ampliadas), d) precisão no conjunto de imagens ampliadas. . . . .	62
Figura A.1–Exemplo de imagens de treinamento:(a) <i>Agrotis ipsilon</i> , (b) <i>Diatraea saccharalis</i> , (c) <i>Helicoverpa zea</i> , (d) <i>Spodoptera frugiperda</i> , (e) <i>Diabrotica speciosa</i> , (f) <i>Dalbulus maidis</i> , (f) imagens sem pragas. . . . .	70

# Lista de Tabelas

Tabela 1 – Referências mais relevantes à construção do conjunto de imagens. . . . .	15
Tabela 2 – Conjunto inicial de imagens de pragas do milho. . . . .	27
Tabela 3 – Total de imagens no conjunto final após as transformações para ampliação. . . . .	28
Tabela 4 – Desempenho médio (%) para cada modelo testado no conjunto de pragas primárias do milho e imagens sem pragas. . . . .	47
Tabela 5 – Desempenho médio (%) para cada modelo testado com todas as pragas e imagens sem pragas. . . . .	52

# Lista de Algoritmos

Algoritmo 1 – Etapas Principais de Treinamento . . . . .	44
Algoritmo 2 – Algoritmo de Varredura de Taxa Inicial . . . . .	46

# Lista de Abreviaturas e Siglas

<i>MIP</i>	Manejo Integrado de Pragas
<i>USDA</i>	<i>United States Department of Agriculture Foreign Agricultural Service</i>
<i>RNC</i>	Redes Neuraonais Convolucionais
<i>DL</i>	<i>Deep Learning</i> - Aprendizagem Profunda
<i>GPU</i>	Unidade de Processamento Gráfico
<i>HOG</i>	histograma de gradiente orientado ( <i>Histogram of Oriented Gradients</i> )
<i>SIFT</i>	Transformação de Recurso Invariante de Escala <i>Scale-Invariant Feature Transform</i>
<i>ReLU</i>	<i>rectified linear unit</i> - unidade de retificação linear
<i>RBF</i>	Função de Base Radial ( <i>radial basis function</i> )
<i>PNG</i>	<i>Portable Network Graphics</i>
<i>RNA</i>	redes neuronais artificiais ( <i>Central Processing Unit</i> )
<i>HSI</i>	<i>Hue, Saturation, Intensity</i>
<i>SVM</i>	<i>Support Vector Machine</i> - Máquina de vetores de suporte
<i>LBP</i>	Padrões Binários Locais ( <i>Local binary patterns</i> )



# Lista de Símbolos

$A$  Símbolo 1 [ $N/m^2$ ]

$\rho$  Símbolo 2 [ $kg/m^3$ ]

# 1 Introdução

O Capítulo 1 apresenta as questões motivacionais concernentes ao manejo do milho, sua relevância nutricional e importância econômica, bem como os impeditivos atuais causados por pragas para uma maior e melhor produção nas lavouras de milho. Tais pragas justificam o problema da construção de um sistema capaz de reconhecer, por meio de uma rede neuronal, as principais pragas de lavouras de milho, através de imagens de campo. Ainda são apresentados os objetivos os quais direcionam as proposições deste trabalho a fim de apresentar uma solução para os problemas em milharais.

## 1.1 Motivação

As plantações de *Zea mays ssp.*, mais conhecido como milho, podem ser encontradas em muitas partes do mundo, contudo, a planta tem suas origens nas Américas, com seus primeiros relatos de domesticação agrícola ocorrido no México, por volta de 9.000 anos atrás (BLAKE, 2015). O milho encontra-se atualmente entre os três cereais mais cultivados em escala mundial (OECD, 2018) conforme destacado na Figura 1.1. Além de ser uma importante fonte de alimento e base alimentar de muitos povos, o milho vigora também como matéria prima na produção de etanol.

Atualmente, países como os Estados Unidos, China e Brasil, são responsáveis respectivamente por cerca de 33,67%, 22,40% e 8,91% da produção mundial (USDA, 2019a), enquanto outros países contribuem com os outros sobejo 35%. Deste modo, a Figura 1.2 ilustra a produção da safra agrícola mundial referente ao ano 2018/19. Nos três países supramencionados, a produção se tornou majoritariamente mecanizada e repleta de inovações tecnológicas (FAOSTAT, 2019). Em 2017, mais de 197 milhões de hectares de área efetivamente colhida foram de milharais (FAOSTAT, 2019).

Contudo, a produção nas lavouras de milho poderia ser ainda maior, caso houvesse o uso efetivo do manejo integrado de pragas, em que as pragas podem causar perdas dentro de um intervalo de 9 a 87% da produção, dependendo da região e da susceptibilidade das plantas ao ataque (SUGAYAMA et al., 2014; CRUZ; VIANA; WAQUIL, 2002; VALICENTE, 2015). Desta forma, a identificação das principais pragas do milho consiste um dos processos fundamentais de coleta e gerenciamento de informações biológicas acerca das espécies residentes nas lavouras. Historicamente, o processo de identificação é baseado

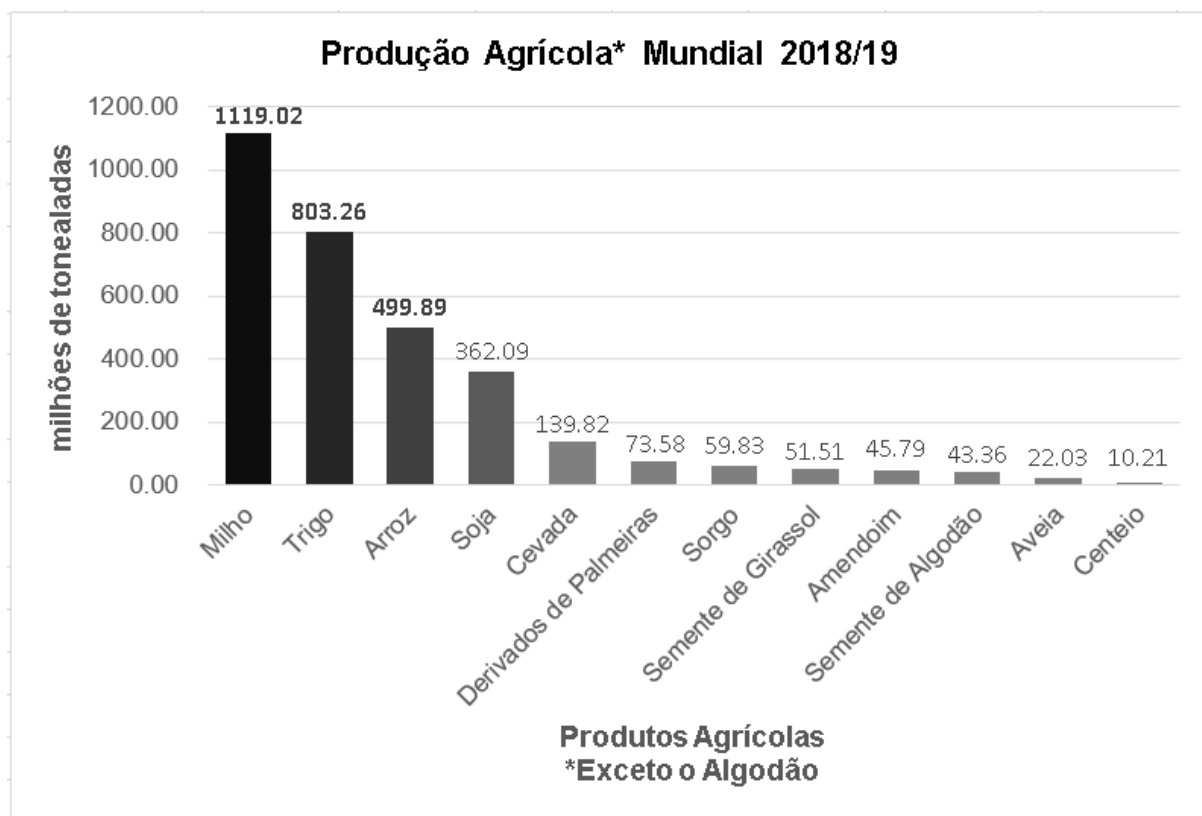


Figura 1.1 – Principais produções agrícolas para a safra de 2018/19 o relatório do Departamento de Agricultura dos Estados Unidos (USDA)(USDA, 2019b), com exceção ao algodão por não por ser mensurado em milhões de libras.

em diagnóstico morfológico fornecido por estudos taxonômicos. Por isso, tal atividade torna-se uma exclusividade de especialistas como taxonomistas e técnicos habilitados a identificar com precisão os insetos, pois requer conhecimento específico adquirido através de uma rica experiência (LIM; KIM; KIM, 2017).

Além disso, os métodos automatizados para a classificação de insetos são essenciais para não-especialistas, em que se requer a construção de um conjunto de imagens de pragas categorizadas. Ressaltando assim, a importância da construção do conjunto de imagens de pragas em lavouras do milho, contudo, para tal feito, constatou-se que na literatura, bem como por buscas por bases específicas na internet, atestando a escassez dos dados de imagens de campo. Desta forma, vê-se a necessidade de se construir tal conjunto de imagens que seja satisfatório ao desenvolvimento de um sistema automatizado de reconhecimento de pragas (LIM; KIM; KIM, 2017).

## 1.2 Problemática

O cultivo do milho está repleto de questões ambientais referentes ao uso sustentável de recursos naturais, como por exemplo, a aplicação de inseticidas no controle de populações de insetos hostis ao milharal e, conseqüentemente, à produção (FAOSTAT, 2019). Os resíduos advindos dos pesticidas aplicados durante as fases do plantio tornam-se, a cada

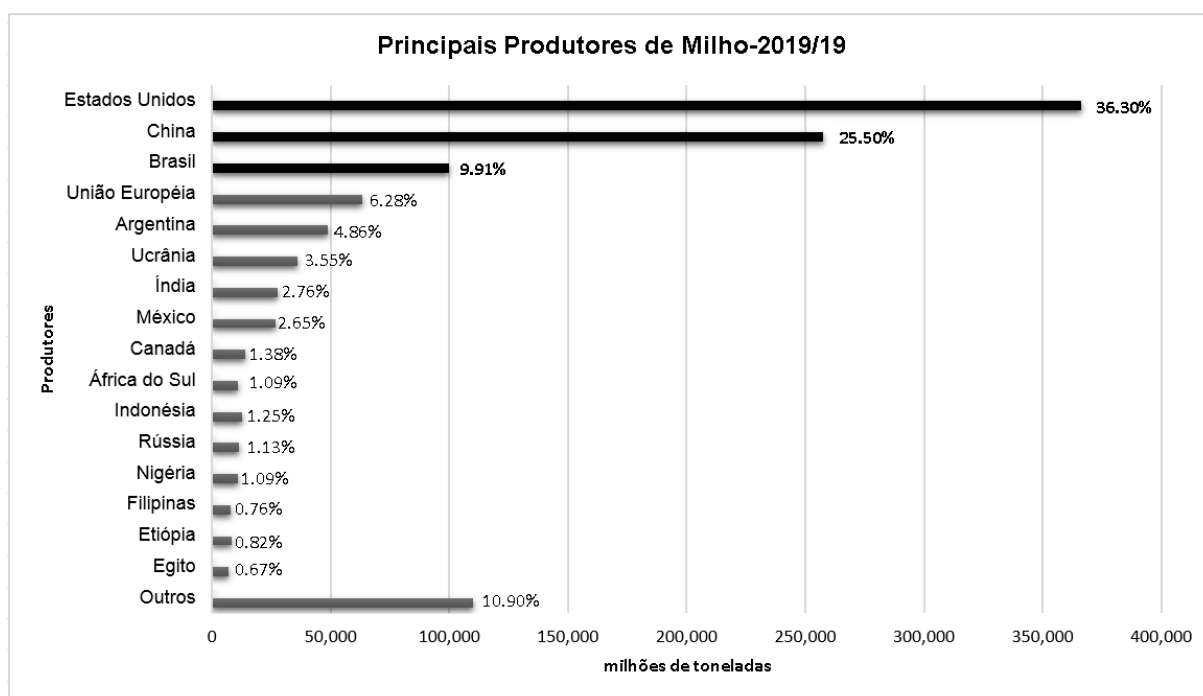


Figura 1.2 – Distribuição mundial da produção de milho, safra 2019 segundo o relatório do USDA (USDA, 2019a).

ano, um fator de risco sanitário, especialmente em países em desenvolvimento devido ao clima quente, mais favorável à presença de mais insetos. A questão sanitária dispõe de leis imprópriamente formuladas quanto ao uso de pesticidas em general, de forma que o real impacto dos resíduos de pesticidas esteja mal dimensionado. Devido a tais implicações, o sistema de manejo integrado de pragas (MIP) (VALICENTE, 2015) pode fornecer estratégias decisivas para suprimir o uso de agentes químicos nocivos à saúde humana aos ecossistemas participantes, onde foram estabelecidas as plantações. Assim, uma das estratégias do MIP para essa questão desafiadora pode ser a construção de uma sistema que utilize uma rede neuronal no reconhecimento das principais pragas presentes na planta, diretamente por imagens tomadas em campo para usufruto posterior em ações de controle dos insetos (CARVALHO, 2017).

O MIP é decisivo na manutenção da saúde das plantações (SCHNEIDER et al., 2015). Algumas das atividades que mais contribuem para o crescimento da produtividade agrícola é o reconhecimento preciso e o controle de pragas. Em contraste a um sistema automático de reconhecimento de pragas (vital para a manutenção da saúde das plantas), sistemas tradicionais utilizados atualmente como o método manual, por exemplo, são lentos e dispendiosos. Felizmente, os avanços nos campos de Visão Computacional e Aprendizagem de Máquina, por outro lado, oferecem soluções promissoras em problemas de reconhecimento de objetos (KAMILARIS; PRENAFETA-BOLDÚ, 2018).

Assim, o problema de concepção de um sistema de reconhecimento por imagens se apoia em avanços na área de Visão Computacional, onde o uso de sistemas de classificação de imagens é proposto, onde cada vez mais é comum o uso de arquiteturas de rede

convolucionais com aprendizagem profunda neste contexto (KAMILARIS; PRENAFETA-BOLDÚ, 2018; LECUN; BENGIO; HINTON, 2015). Martineau (MARTINEAU et al., 2017) sintetizaram os 44 (quarenta e quatro) trabalhos mais recentes de estudos de reconhecimento de insetos através de imagens. Em sua maioria, as imagens dos estudos são fotografias tomadas em laboratório e sob condições restritas. Apenas três deles, os quais representam aproximadamente 7% das propostas, foram desenvolvidas com imagens de campo. Ademais, as circunstâncias de aquisição são outro fator preponderante, visto que tais circunstâncias são diversas como se vê na Figura 1.3. Fotografias de insetos tomadas em campo possuem um segundo plano com alta variabilidade de formas, iluminação, além da constante mudança de posição dos insetos nas plantas, no ar e no solo, ou seja, condições ausentes em comparação com imagens de laboratório as quais têm plano de fundo de fundo simplificado.

### 1.3 Objetivos

Uma vez definido o problema de construção de um sistema de redes neuronais convolucionais para o reconhecimento de pragas do milho em imagens de campo, este trabalho tem como objetivos gerais:

- Criar um conjunto de imagens de campo constituído das pragas primárias e secundárias presentes em lavouras de milho;
- Aferir a exatidão de redes neuronais no reconhecimento de pragas de milho.

Portanto, necessariamente devem compor os objetivos específicos:

- Comparar arquiteturas de redes neuronais clássicas de reconhecimento de imagem utilizadas na agricultura com novos classificadores basados em aprendizagem profunda.
- Propor uma variação de uma rede neuronal convolucional profunda (Inception-v3) para o problema de reconhecimento de pragas do milho.

### 1.4 Metodologia

O procedimento deste trabalho é constituído de três grandes fases, ilustradas na Figura 1.4 i) criação do conjunto de imagens, ii) desenvolvimento e aplicação do sistema de reconhecimento de imagens de pragas e iii) avaliação do desempenho das redes por meio de métricas estatísticas.

Assim, os principais detalhes de cada fase são como se segue:

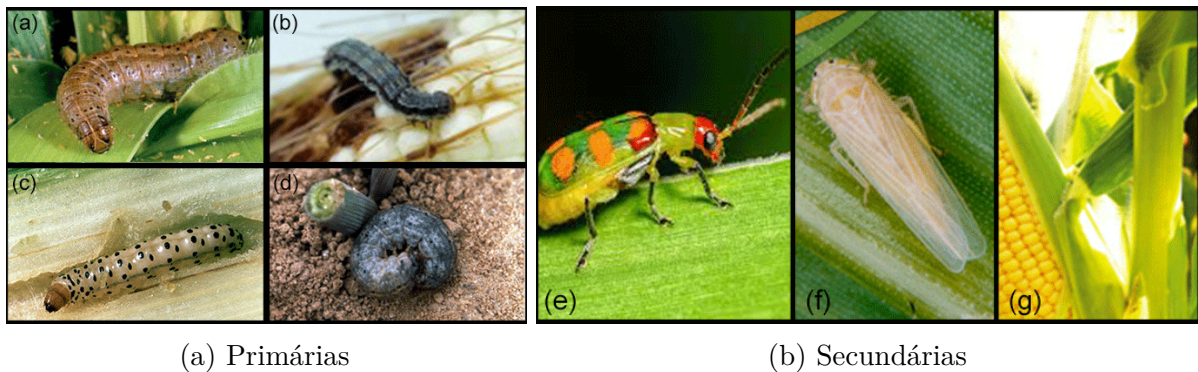


Figura 1.3 – Amostras de pragas do milho nas circunstâncias encontradas durante a aquisição das imagens em campo aberto, I) Pragas Primárias: a) *Spodoptera frugiperda*, b) *Helicoverpa zea*, c) *Diatraea saccharalis*, d) *Agrotis ipsilon*; II) Secundárias e) *Diabrotica speciosa*, f) *Dalbulus maidis*, g) Imagens de milharais sem pragas. Todas as imagens de treinamento utilizadas neste trabalho encontram-se disponíveis no Apêndice A.



Figura 1.4 – Fluxograma das três grandes fases do trabalho.

- **Criação do conjunto de imagens:** É constituído de um inédito conjunto de imagens de pragas de campo, originais (700) e ampliadas (4340), no qual contempla-se das 7 principais pragas do milho.
- **Desenvolvimento e aplicação:** integração do conjunto de dados para realização do treinamento e registro das 4 redes neuronais implementadas: uma rede clássica, uma rede convolutiva, uma rede convolutiva residual e uma proposta de variação da rede convolutiva.
- **Avaliação:** por fim, os resultados de cada rede foram comparados de acordo com os métodos de precisão por uma validação cruzada de 8 iterações e F-score.

## 1.5 Organização do Trabalho

Apresentados os principais fatores motivantes para a realização deste trabalho, os capítulos seguem desta forma:

O Capítulo 2 descreve cada uma das pragas do milho as quais foram utilizadas na construção do conjunto de imagens. Em seguida, são apresentadas as propostas referentes a reconhecimento de pragas por imagens em lavouras.

Em seguida, o Capítulo 3 aborda os fundamentos necessários para a criação do conjunto de imagens como o pré-processamento de imagens. Neste capítulo, ainda são introduzidas as redes neuronais (clássica, convolutiva e residual) implementadas, bem como suas funções e utilidade de cada uma de suas camadas. Também neste capítulo encontram-se as modificações contidas na proposta de modificação à rede residual Inception-V3.

O Capítulo 4 descreve os passos pertinentes à construção do conjunto de imagens. A segunda seção do capítulo aborda todas as etapas de treinamento de cada rede neuronal utilizada no trabalho.

Por ordem, o Capítulo 5 os resultados, tanto da criação do conjunto de imagens segundo os critérios descritos no capítulo 4 e os resultados para as redes neuronais testadas.

O Capítulo 6 descreve as considerações produzidas após o término dos experimentos realizados. Ainda considera-se quais são os requisitos necessários para se realizar trabalhos futuros.

O Apêndice A contém figuras em miniatura de cada uma das pragas utilizadas durante o experimento.

## 2 Revisão da Literatura

O Capítulo 2 descreve as seis principais pragas do milho (quatro primárias e duas secundárias). Também neste capítulo está descrito o estado da arte em reconhecimento de pragas em lavouras em seus diversos estágios e ambientes de captura.

### 2.1 Pragas Primárias e Secundárias na Cultura do Milho

A diversos procedimentos em classificar as pragas de milho como primárias e secundárias, dependendo da importância do dano que causam na cultura. Mas, em razão da importância relativa de determinadas pragas em determinadas regiões, serão destacadas pragas de importância econômica e/ou de maior ocorrência na cultura do milho, a figura 2.1 classifica quais são as classes de pragas pertencentes a categoria de pragas primárias e secundárias bem como os diferentes estágios de ataque.

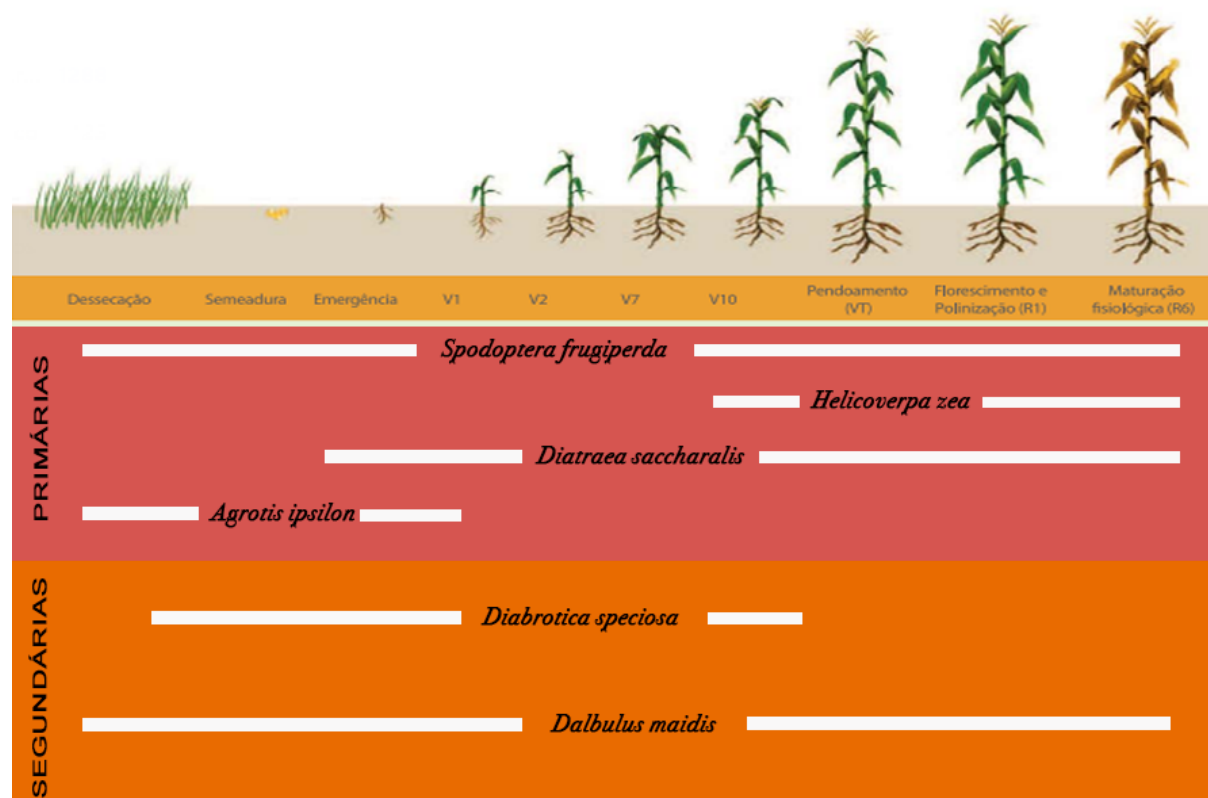


Figura 2.1 – Principais pragas do milho em seus diferentes estágios.



Espécies de determinados insetos, ácaros, entre outros pequenos animais, podem ser naturalmente encontrados em quantidades controladas, em seus ecossistemas de habitação (MEDEIROS, 2010). Entretanto, ao substituir-se tal ecossistema, por vezes é bastante complexo, por um sistema simplificado (de domínio de uma única planta) a quantidade de alguns desses pequenos animais pode ser favorecida pelo novo ecossistema, constituindo nesta conjuntura, uma praga ao sistema simplificado (MEDEIROS, 2010).

O controle de pragas na cultura do milho é indispensável, devido ao potencial destrutivo das pragas em qualquer estágio que se encontrem (SCHNEIDER et al., 2015; MOREIRA; ARAGÃO, 2009). Além de prejudicar o desenvolvimento da planta, de forma a impedir posteriormente o desenvolvimento pleno do fruto, a eclosão de pragas também reduz a população de plantas por área plantada (SCHNEIDER et al., 2015). Assim, estratégias de controle e combate às ameaças à lavouras são parte do MIP o qual essencialmente estabelece o equilíbrio entre os danos oriundos das populações prejudiciais à lavoura e as estratégias de controle com base no nível de infestação (SCHNEIDER et al., 2015).

O monitoramento é o principal recurso necessário para a aplicação do MIP, por outro lado, este está alicerçado em aplicações de inseticidas, além de recomendações no uso de tecnologias genéticas e culturas rotativas (SCHNEIDER et al., 2015). O monitoramento, entretanto, é conduzido através de visitas periódicas às lavouras (MOREIRA; ARAGÃO, 2009) onde especialistas observam a frequências das populações, o estágio de desenvolvimento, condições climáticas para determinar quais pragas estão em seu período favorável de proliferação (MOREIRA; ARAGÃO, 2009) .

Na cultura do milho, as pragas podem ser agrupadas de acordo com a região de ataque (MOREIRA; ARAGÃO, 2009) e também de acordo com estágio de desenvolvimento da planta (SCHNEIDER et al., 2015). Estas regiões seguem a localização das partes da planta: raiz, colmo, folhas e espiga (MOREIRA; ARAGÃO, 2009). Ainda que divididas em pragas primárias e secundárias, é possível observar a distribuição das mesmas segundo a combinação dos critérios de localização e estágio de desenvolvimento da planta (SCHNEIDER et al., 2015). De acordo com (SCHNEIDER et al., 2015), o desenvolvimento do milho pode ser disposto em dez fases, desde a dessecação das ervas da área de plantio até a fase final de maturação do planta em ponto de colheita.

A divisão entre pragas primárias e secundárias dá-se através do grau de destruição e prejuízo ao desenvolvimento ocasionado por determinada população.

### 2.1.1 Pragas Primárias

Segundo (MOREIRA; ARAGÃO, 2009), nas lavouras de milho, pelo menos 31 espécies de pragas podem ser encontradas, dentre as quais (SCHNEIDER et al., 2015) classifica cinco delas como primárias (*Spodoptera frugiperda*, *Helicoverpa zea*, *Diatraea saccharalis*, *Elasmopalpus lignosellus* e *Agrotis ipsilon*) e outras oito espécies como secundá-

rias (*Diabrotica speciosa*, *Conoderus scalaris*, *Anomala testaceipennis*, *Astylus variegatus*, *Dalbulus maidis*, *Alabama argillacea*, cupins e um grupo de percevejos os quais variam de acordo com a região e rotação de culturas). Dentre as primárias, não compõem o objeto de estudo, devido a sua baixa frequência, de amostras, *Elasmopalpus lignosellus*. Pelo mesmo princípio de frequência, durante o estudo as secundárias observadas foram *Diabrotica speciosa* e *Dalbulus maidis*.

#### 2.1.1.1 *Spodoptera frugiperda*

A espécie *Spodoptera frugiperda* é uma praga de migração, pois sua fonte de alimentos distribui-se por mais de 80 tipos de plantas dentre as quais o milho é considerado um dos principais prejudicados com a população da espécie. O ataque ocorre principalmente nas fases iniciais da planta e no desenvolvimento dos grãos nas espigas. Ademais, a *S. frugiperda* se favorece através da cultura rotativa durante as estações de inverno e verão (SCHNEIDER et al., 2015).

O ciclo biológico desta praga vai desde sua fase larval seguida por sua transfiguração em mariposa, onde seus hábitos alimentares concentram-se no período noturno. Os hábitos alimentares antropófagos não permitem que haja ocorrência de mais de uma lagarta por planta (MOREIRA; ARAGÃO, 2009). Neste trabalho, considerou-se apenas o seu estado larval e mais destrutivo, Figura 2.2, onde a lagarta transita por partes da planta para se alimentar.



Figura 2.2 – Amostras de *Spodoptera frugiperda* encontrada em diferentes partes da planta. cada imagem foi recortada em formato quadrado de 224 x 224.

#### 2.1.1.2 *Agrotis ipsilon*

A ocorrência predominantemente terrestre da espécie *Agrotis ipsilon* coloca-a como uma das principais ameaças à planta em sua fase de emergência, quando presente na plantação (SCHNEIDER et al., 2015). Felizmente, o ciclo da praga perdura entre a dessecação e os primeiros centímetros de altura acima do solo, ou seja em torno de trinta dias (MOREIRA; ARAGÃO, 2009). Além da observância da própria lagarta, os indícios de seus ataques são caracterizados por plantas jovens tombadas e plantadas adultas com perfurações e túneis iniciados a partir da base, próxima ao solo, Figura 2.3.



Figura 2.3 – Amostras de *Agrotis ipsilon* em localizações no solo próximas à planta.

#### 2.1.1.3 *Diatraea saccharalis*

As lagartas da espécie vistas na Figura 2.4 podem variar de comprimento (entre 22 e 25mm) e fenótipo específico como cabeça marrom, coloração corporal de variação entre cores branca e amarela e distribuição de pontos pretos ao longo do seu dorso (SCHNEIDER et al., 2015; MOREIRA; ARAGÃO, 2009).



Figura 2.4 – Amostras de *Diatraea saccharalis* em localizações frequentes e os danos que estas podem impor à planta.

Inicialmente, as lagartas causam apenas uma leve raspagem sobre as folhas por serem minúsculas (2mm de comprimento), (MOREIRA; ARAGÃO, 2009). Entretanto, à medida em que aproximam-se de seu larval máximo, estas alimentam-se das espigas e caule, formando uma galeria de túneis na planta. Este túneis servem de depósito para seus casulos por meio dos quais, as lagartas preparam-se para a fase como mariposa (MOREIRA; ARAGÃO, 2009). Assim, *D. saccharalis* torna-se uma das pragas mais danosas à planta devido ao seu ciclo biológico durar tanto quanto o ciclo da planta (SCHNEIDER et al., 2015).

#### 2.1.1.4 *Helicoverpa zea*

Uma lagarta de características visuais próximas a *Spodoptera frugiperda*, a *Helicoverpa zea*, Figura 2.5, encontra nos cabelos do milho e nos grãos no topo das espigas, a sua fonte de alimentos (MOREIRA; ARAGÃO, 2009). No auge de seu desenvolvimento larval, a *H. zea* chega a 5cm de comprimento. A formação de casulos (pupas) para a metamorfose é precedida por sua migração para o solo até 25 cm de profundidade, de onde emergirá para a sua segunda fase como mariposa. Não somente as lagartas causam danos ao fruto e ao colmo da planta, mas também as cavidades abertas atraem outros insetos e causam



Figura 2.5 – Amostras de *Helicoverpa zea* nas pontas das espigas, principal região de ataque praga nas plantas.

apodrecimento na região. A maior atividade de ataque da *H. zea* compreende desde antes a formação de pendões até a maturação do milho.

## 2.1.2 Pragas Secundárias

### 2.1.2.1 *Dalbulus maidis*

Um pequeno inseto alado, predominantemente monotônico com pequena variação de cor principal entre o branco, tons de amarelo, verde e marrom, *Dalbulus maidis* é uma praga de hábitos capaz de transmitir fitopatógenos, seu principal dano às plantas (SCHNEIDER et al., 2015; MOREIRA; ARAGÃO, 2009). Esta praga de menos de 5mm de comprimento, Figura 2.6, é sugadora, pois suas ninfas e seus adultos alimenta-se da seiva das plantas (SCHNEIDER et al., 2015). Desta forma, sua saliva contamina planta através da sucção da seiva, causando deformação das folhas e no fruto (MOREIRA; ARAGÃO, 2009). Ademais, sua presença é observável durante todas as fases do cultivo em suas colônias, desde a dessecação até a fase de colheita (SCHNEIDER et al., 2015).



Figura 2.6 – Amostras de *Dalbulus maidis* em localizações no solo próximas à planta.

### 2.1.2.2 *Diabrotica speciosa*

*Diabrotica speciosa*, Figura 2.7, é um besouro de até 6mm em sua fase adulta e de coloração verde e manchas ovais amarelas (MOREIRA; ARAGÃO, 2009). Suas larvas penetram o caule, impedindo o fluxo da seiva para as partes aéreas da planta. Em fases iniciais, o bloqueio no caule impede o crescimento apropriado, já na fase adulta da planta, a presença de larvas de *D. speciosa* no caule pode provocar desde a interrupção do pleno desenvolvimento do fruto até mesmo o tombamento da planta (SCHNEIDER et al., 2015).

Na presença de outras pragas como o *Spodoptera frugiperda*, as perfurações abertas pela *D. speciosa* podem ser atribuídas àquela outra.

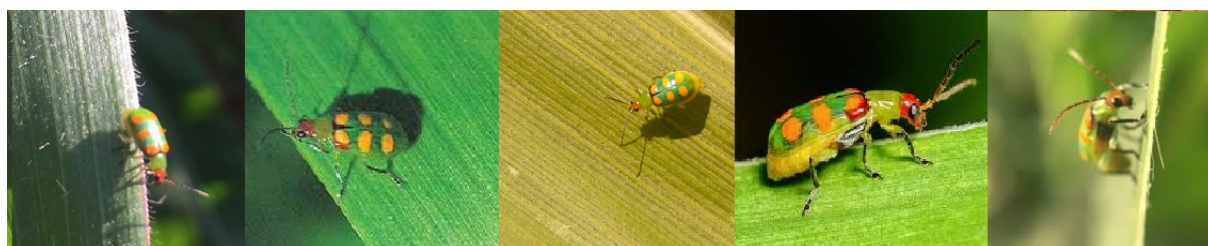


Figura 2.7 – Amostras de *Diabrotica speciosa* em localizações no solo próximas à planta.

### 2.1.2.3 Outras pragas secundárias: Percevejos

Algumas pragas secundárias, segundo (SCHNEIDER et al., 2015) são avaliadas atualmente por cada região e histórico, por ocorrer em menor frequência e quantidade e com menor dano quando comparado às outras pragas primárias e secundárias. Por exemplo, o grupo dos percevejos (SCHNEIDER et al., 2015) até o momento enquadra-se no critério de avaliação; estes são compostos pelas seguintes espécies: *Leptoglossus zonatus*, *Dichelops melachanthus*, *Dichelops furcatus* e *Nezara viridula*.

## 2.2 Estado da Arte em Reconhecimento de Pragas por Imagem

A identificação de pragas vivas é desafiadora e especialmente relevante para as plantações. Assim, os taxonomistas têm árdua tarefas de continuamente capturar e classificar insetos e outros animais em campos agrícolas. No uso de métodos computacionais, comumente encontra-se três ambientes: laboratório (as imagens dos animais são tomadas com fundos de cor sólida e posição fixa)(MARTINEAU et al., 2017), plantações em estufas (MAHARLOOEI et al., 2017; ESPINOZA et al., 2016; MUNDADA; GOHOKAR, 2013) e em campos (LIU et al., 2016; MARTINEAU et al., 2017). Em sua maior composição, as tentativas iniciais de reconhecimento partiram do pré-processamento e segmentação dos alvos, separando-os de outros elementos presentes nas imagens adquiridas(MAHARLOOEI et al., 2017).

No ambiente de uma estufa, por exemplo, descrito por (ESPINOZA et al., 2016) para realizar a tarefa de reconhecimento de *Bemisia tabaci* e *Frankliniella occidentalis* utilizou-se 28 configurações diferentes de redes neurais artificiais (RNA). Os insetos foram capturados tanto em papeis pega-mosca de cor azul amarelo ou azul, em seguida realizou-se a segmentação dos insetos do plano de fundo através de análise de histograma. Para o teste, alcançou-se valores de *F-score* de 0,91 para ambos os insetos. Em (EBRAHIMI et al., 2017), classificou-se os insetos em estufas de morango por meio de SVM (*support vector machine* - máquina de suporte de vetores) com extração por RBF (*radial basis*

*function*- função de base radial) combinado com pré-processamento HSI (*Hue, Saturation, Intensity*) e L/W para as características de cores. No estudo, tripes em fases larval e adulta foram considerados as maiores pragas. Ademais, todas as imagens foram segmentadas do plano de fundo por um operador gama e reconhecimento alimentado pelos pixels das áreas segmentadas.

De acordo com o levantamento em (MARTINEAU et al., 2017), somente três das quarenta e quatro pesquisas conduzidas foram em campo aberto, um total de 7% da publicação revisada em reconhecimento de pragas. Em (MAYO; WATSON, 2006) foram coletadas 774 imagens de 35 espécies de traças vivas, capturadas no jardim botânico e fotografadas sobre um fundo branco, de fácil identificação. Também foram utilizadas características como densidade e angulação e área de preenchimento. Em (FAITHPRAISE et al., 2013), as imagens em campo aberto utilizadas no experimento tiveram segmentação em cada canal realizada por agrupamentos os quais foram identificados no processo. Assim, o procedimento buscou separar os insetos de folhas esverdeadas, por meio de segmentação *k-means*. Posteriormente no experimento, um filtro de correspondência foi treinado com variações das imagens geradas por transformação. Assim a correspondência obtida e combinada com um limiar foram utilizadas nas imagens de entrada. Segundo (FAITHPRAISE et al., 2013), houve êxito na aplicação do método para 30 tipo de plantações.

Outra proposta tentou codificar de forma esparsa 24 classes de várias culturas ao extrair um conjunto de características como forma cor, textura, SIFT e HOG (XIE et al., 2015). Neste método foi utilizado um conjunto de filtros na aprendizagem de em imagens com enquadramento próximo, onde os insetos ocupavam 85-90% do quadro. O conjunto de dados seguiu a proporção de 60:25 de treino para validação, alcançando 85,5% no conjunto. Os avanços no estado-da-arte de reconhecimento de objetos (LECUN; BENGIO; HINTON, 2015) proporcionou o surgimento de novos modelos de aprendizagem com múltiplas camadas e habilidade reconhecimento diretamente das imagens em menor tempo e sem necessidade de modelagem de características. Assim a aprendizagem profunda pode ser recentemente aplicada também na agricultura (KAMILARIS; PRENAFETA-BOLDÚ, 2018). Aplicações agrícolas com redes neuronais profundas cuidam de reconhecimento de doenças em folhas, reconhecimento de plantas, além de estimativas de produção durante as safras (KAMILARIS; PRENAFETA-BOLDÚ, 2018). Semelhantemente, redes convolucionais profundas também têm sido empregadas em reconhecimento de pragas (LIM; KIM; KIM, 2017; CHENG et al., 2017).

A arquitetura de rede convolucional típica, AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), empregada em (LIM; KIM; KIM, 2017) foi utilizada para reconhecer 27 espécie de insetos do banco de imagens *ImageNet* (DENG et al., 2009). Os resultados revelaram que os parâmetros do conjunto de imagens como centralização de objeto e densidade são algumas das características contribuintes para o desempenho final do modelo de exatidão média de 81.82% no experimento.

Existem, no entanto, condições especiais quanto ao reconhecimento de pragas. Muitos insetos em plantações específicas, são em sua maioria, inofensivos ou mesmo parte da defesa biológica, enquanto outros são majoritariamente pragas prejudiciais a diversas culturas. Em culturas de soja e milho, por exemplo, indivíduos de mesma espécie podem apresentar características diferentes, de acordo com a cultura atual sob ataque. Podem vir a ocorrer, em tais condições, mutações a quais exigem maior sensibilidade dos modelos de reconhecimento.

Um passo essencial a se dar antes de tudo é avaliar qual cultura será observada e quais são as populações locais residentes, tanto inofensivas quanto pragas primárias e secundárias. Em (XIE et al., 2015), abordou-se a classificação de insetos com uso de imagens de campo aberto e modelos de aprendizagem profunda. No estudo mencionado, 10 espécies de insetos com 55 imagens cada, foram selecionadas para compor o conjunto de teste. O modelo principal de aprendizagem, uma rede convolucional típica (AlexNet) foi posta a prova contra duas redes de arquitetura residual (ResNet50 e ResNet101), alcançando respectivamente de 86,67%, 94,67% e 98,67%, resultados expressivos de exatidão no conjunto testado. Neste caso, os insetos eram significativamente distintos entre si bem como oriundos de plantações de culturas distintas.

A avaliação de pragas a partir de imagens de campo, distribuídos segundo o nível de danos - primária e secundária - em determinado ecossistema simplificado (lavoura), possui inúmeros desafios a serem abordados, pois tratam-se de um recorte da realidade com diversas implicações quanto ao reconhecimento de pragas.

Portanto, observa-se que os trabalhos mais relevante à fundamentação deste abordam principalmente redes neuronais clássicas como SVM, rede convolucionais típicas com grafos em cadeia direta, por exemplo, AlexNet e mais recentemente, redes convolucionais residuais. Além disso, verifica-se a ocorrência de trabalhos os quais abordam a identificação de insetos em ecossistemas diversos, ou seja, aqueles cuja relação não por meio da ocorrência em um cultivo específico como se observa na Tabela 1. Há também propostas de identificação de insetos considerados pragas para determinada cultura, como a do morango observados em estufas. Ainda, observa-se a ausência de trabalhos especificamente relacionados aos insetos e outros animais considerados como pragas às lavouras do milho, onde um apenas tratou de abordar doenças foliares do milho sem insetos.

Diante disso, este trabalho aborda, por conseguinte, o reconhecimento de pragas do milho por meio de redes neuronais. Para a execução da tarefa, é necessário a construção de um conjunto de imagens, neste caso, imagens de campo da pragas do milho. Tal conjunto, exclusivamente composto por imagens de pragas do milho, deixa de estar indisponível a partir deste trabalho, uma que propôs-se a realizar a tarefa de construção do conjunto de imagens de pragas do milho, tomadas em campo aberto.

Uma vez disponível o conjunto de imagens de pragas do milho, torna-se possível a aplicação de arquiteturas de base de redes neuronais para que se realize o comparativo

Trabalho	Lab	Estufa	Campo	Plantação
(MAYO; WATSON, 2006)	×	-	-	diversa
(ESPINOZA et al., 2016)	×	×	-	tomate
(MAHARLOOEI et al., 2017)	-	×	-	soja
(MUNDADA; GOHOKAR, 2013)	-	×	-	flores/hortaliças
(EBRAHIMI et al., 2017)	-	×	-	morango
(FAITHPRAISE et al., 2013)	-	-	×	diversa
(XIE et al., 2015)	-	-	×	diversa
(LIU et al., 2016)	-	-	×	diversa

Tabela 1 – Referências mais relevantes à construção do conjunto de imagens.

no desempenho de classificação das classes do conjunto. A estas arquiteturas de base, adicionou-se também uma modificação proposta de uma rede convolucional residual. Desta forma, as arquiteturas de base possuem três modelos: uma rede neuronal clássica (SVM-LBP), uma rede convolucional direta ou típica (AlexNet) e uma rede convolucional residual, Inception-V3 (SZEGEDY et al., 2016). A rede modificada proposta neste trabalho é baseada no modelo original Inception-V3.

Entretanto, para melhor compreensão do funcionamento de cada arquitetura, necessita-se verificar antes quais são as principais funções elementares utilizadas na construção de cada arquitetura, tanto das neuronais clássicas, quanto das convolucionais típicas e das residuais.

## 2.3 Redes Neurais Artificiais

Tarefas de reconhecimento e classificação de objetos em Visão Computacional vem melhorando sua exatidão a cada ano através do uso de redes neuronais convolucionais (RNC) e aprendizagem profunda (DL - *Deep Learning*) (LECUN; BENGIO; HINTON, 2015). Grande parte das RNCs são projetadas com sequências de camadas distintas de acordo com seu propósito de uso. Dentre as camadas mais comuns estão as que têm função de regularização, filtros convolucionais, unidades lineares retificadas (ReLU- *rectified linear unit*) (NAIR; HINTON, 2010), reorganização de camadas em vetores e funções preditivas como sigmoid a qual é comumente utilizada em problemas de variáveis contínuas e *softmax* para casos de problemas de rotulação. Além da arquitetura, a possibilidade de herdar características por meio de treinamentos realizados anteriormente com outros conjuntos de imagens, faz com que tarefas de alta complexidade passíveis de se realizar com alto grau de confiança quando aplicados a novos conjuntos de dados, obtidos de outras aplicações (LECUN; BENGIO; HINTON, 2015).



### 2.3.1 Redes Neurais Clássicas em Aprendizagem de Máquina

A aprendizagem de máquina tem sido aplicada em três categorias distintas: aprendizagem supervisionada, não-supervisionada (GOODFELLOW; BENGIO; COURVILLE, 2016) e a mais recente trata de geração de novos dados similares à distribuição aos de entrada da rede (GOODFELLOW et al., 2014). Isto pode ser posto em contraste com as duas categorias iniciais as quais buscam classificar ou prever situações a partir de dados de entrada.

Em problemas envolvendo aprendizagem supervisionada, a previsão pode fornecer dados dentro de um conjunto teoricamente infinito de possibilidades, também conhecido como problema de regressão. O segundo problema abordado em aprendizagem supervisionada, prevê saídas dentro um conjunto finito de categorias, por exemplo, 0 ou 1. Nos dois casos, conhece-se, durante o treinamento, tanto as entradas quanto as saídas e, desta forma tem-se um conjunto finito de pares de entradas e saídas. Quando tal conjunto é insuficiente, deve-se antes coletar mais pares para compor conjunto ou até mesmo lançar mão de técnicas que supram tal carência de dados.

Na aprendizagem não-supervisionada, não há pares de entradas e saídas previamente conhecidas. Antes, estabelece-se um cenário com um objetivo definido a ser alcançado. Assim, o algoritmo empregado, deve extrair estruturas do cenário de dados e tomar decisões que permitam chegar ao objetivo final. Por este motivo, não há um conjunto finito para realização do treinamento, o que permite uma tentativa contínua até alcançar o objetivo final, se possível.

#### 2.3.1.1 Estrutura fundamental de uma rede neuronal artificial

Ao considerar a presença de pelo menos, uma camada intermediária, uma rede neuronal conterá pelo menos: uma camada de entrada, uma camada intermediária e camada de saída. Primeiramente, a camada de entrada pode ser expressa por um conjunto de características que represente o dado de entrada, assim, a quantidade de tais características pode ser descrito como  $U_j$  na camada  $j$ . Semelhantemente a camada subsequente  $j + 1$  terá  $U_{j+1}$  unidades. Desta forma, a dimensão da matriz de parâmetros ( $\theta_j$ ) da camada  $j$  para a camada  $j + 1$  pode ser expresso pelo número de unidades da camada  $j + 1$  multiplicada pelo número de camadas da camada  $j$  acrescida de uma unidade ( $\theta_j = U_{j+1} \times U_j + 1$ ). Esse requisito permite a extensão de uma rede para quantas camadas forem possíveis. A Figura 2.8 ilustra todos os elementos básicos de uma rede neuronal: entrada, camada intermediária e camada de predição.

Para um problema de reconhecimento de imagens, por exemplo, as características da camada entrada  $U_j$  da Figura 2.8 podem receber os próprios pixels da imagem a ser classificada, em seguida, os valores de entrada são combinados por um produto escalar de matrizes com os parâmetros de  $\theta_j$ , de forma a produzir os parâmetros da camada oculta

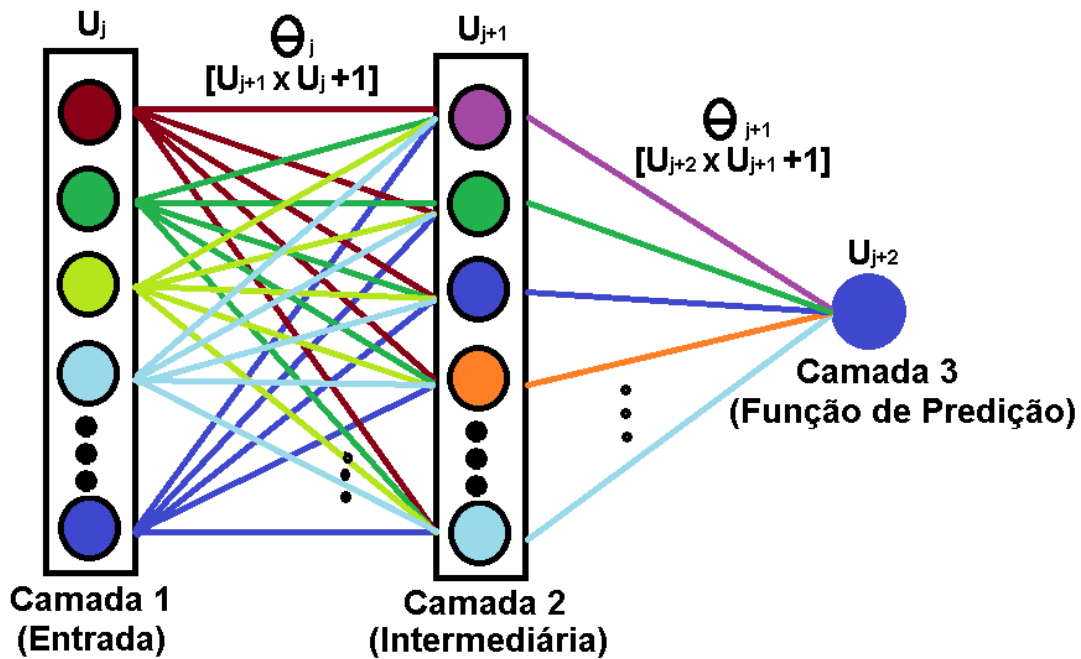


Figura 2.8 – Estrutura fundamental em uma rede neuronal artificial.

$U_{j+1}$ . Os parâmetros da camada oculta, por sua vez, são também combinados por produto escalar com a matriz de parâmetros  $\theta_{j+1}$ , dando origem aos parâmetros utilizados na função de ativação a qual dá origem aos parâmetros em  $U_{j+2}$ , ou seja, a camada de saída responsável por indicar a qual classe pertence a imagem de entrada.

### 2.3.2 Redes Neurais Convolucionais

As redes neurais convolucionais (RNC) são um tipo especializado de modelo de rede neural projetado para trabalhar com dados de imagem sejam elas bidimensionais (imagens de um canal - escala de cinza) ou tridimensionais (imagens coloridas - três canais: vermelho, verde e azul), embora possam ser usados com dados unidimensionais. Ainda, é uma das principais categorias de redes que trabalha com o reconhecimento de imagens, detecção de objetos, como reconhecimento facial, segmentação, entre outras áreas.

Assim como em redes neurais clássicas, a classificação de imagem em RNC é realizada de forma em que se tem uma imagem de entrada, na qual se aplica uma série de camadas de processamento (em sua maior parte, camadas de convolução), em seguida é gerada a classificação da imagem, ou seja, especificar a qual subgrupo do conjunto pertence a imagem de entrada. No processo, os computadores tratam a imagem de entrada como matriz de pixels, a qual depende da resolução da imagem. Com base tal parâmetro de resolução, ter-se-á  $h \times w \times d$  ( $h$  = altura,  $w$  = largura,  $d$  = dimensão). Por exemplo, uma imagem de  $6 \times 6 \times 3$  pode ser representada por uma matriz em que o valor 3 refere-se aos três canais da imagem RGB ( $R$  - red, vermelho;  $G$  - green, verde; e  $blue$ , azul), como visto na Figura 2.9, caso a imagem fosse  $6 \times 6 \times 1$ , esta seria uma imagem em escala de cinza,

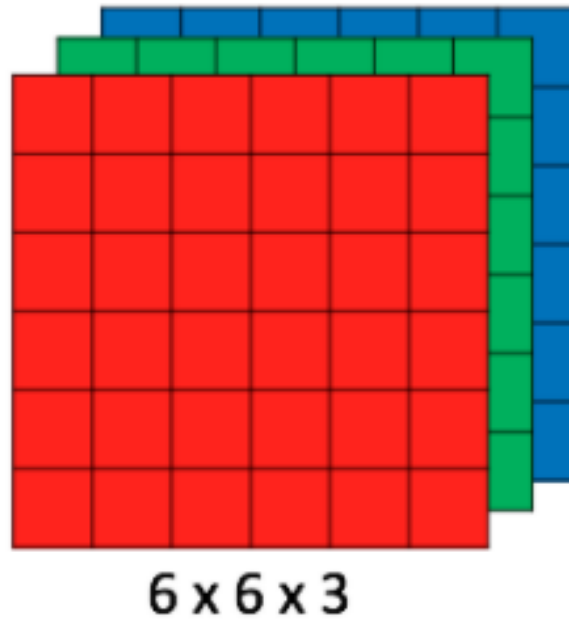


Figura 2.9 – Representação de uma matriz RGB cuja profundidade representa os três canais de um imagem colorida.

representada por uma matriz em que o valor 1 remete-se ao seu único canal.

Além disso, os valores em cada dimensão podem ser determinados por uma série de representações. Para se definir a representação do valor de um pixel é preciso considerar o intervalo de resolução, o número de amostras ou amostragem, e a quantização, ou seja para qual valor representativo o valor real será mapeado (GONZALEZ; WOODS, 2000). Para fins exemplificativos e de acordo com(GONZALEZ; WOODS, 2000), um intervalo de resolução a ser adotado, por exemplo, poder construído a partir de uma base binária. Um intervalo de resolução binária  $L = 8$  compreende valores de intensidade de pixel  $[0, 255]$ , ou seja  $2^8 - 1$  possíveis valores para cada pixel, em que 0 e 255 são os valores mínimo e máximo que um pixel pode assumir. A menor resolução utilizada tem  $L = 1$ , onde cada pixel pode assumir valor 0 ou 1. Na Figura 2.10 tem-se um exemplo de uma imagem a qual teve seus valores de pixel mapeados para resolução  $L = 1$  ou seja, valores de intensidade de pixel no intervalo  $[0, 1]$ , onde o valor máximo é definido por  $2^1 - 1$ . De forma geral, os parâmetros de aquisição de imagem como amostragem, resolução e quantização, bem como a forma de representação, seja RGB, HSI, etc, são definidos no equipamento de captura (GONZALEZ; WOODS, 2000).

### 2.3.2.1 Camada de Convolução

Uma convolução é uma operação linear que envolve a multiplicação de um conjunto de pesos com a entrada, dado que a operação foi projetada para entrada do tipo imagem, por exemplo bidimensional, a multiplicação é realizada entre uma matriz de dados de entrada e uma matriz bidimensional de pesos, chamada de filtro ou *kernel* (KALCHBRENNER;

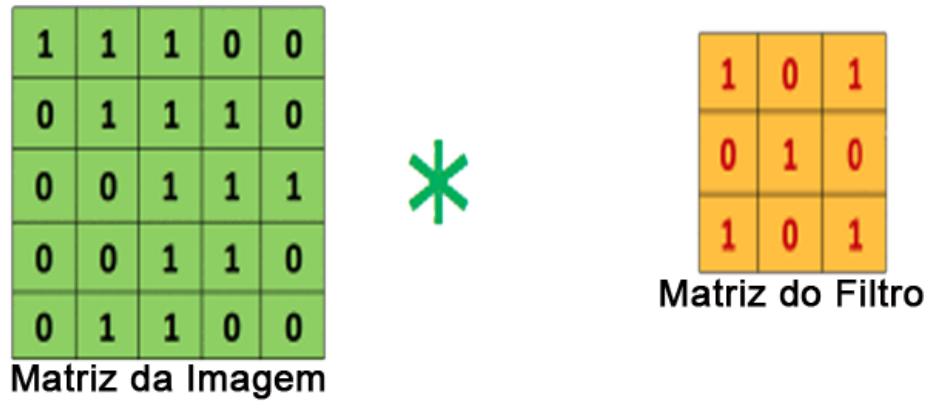


Figura 2.10 – Imagem ilustrativa de uma Matriz de imagem multiplicando kernel ou matriz de filtro

(GREFENSTETTE; BLUNSOM, 2014) conforme a equação 2.1.

$$f(x, y) * g(x, y) = \sum_{m=1}^M \sum_{n=1}^N f(x, y).g(x - m, y - n) \quad (2.1)$$

A convolução é uma operação que se mostrou capaz a extrair características de uma imagem (LECUN; BENGIO; HINTON, 2015), como formas geométricas simples (linhas horizontais, verticais e diagonais), a combinação delas em formas mais complexas, além de texturas e gradientes de cor. A convolução destaca as relações entre os pixels, de forma que estas relações seja organizadas na saída da operação. O filtro é menor que os dados de entrada e o tipo de multiplicação aplicado entre um *patch* do tamanho do filtro da entrada, e o filtro é um produto escalar. Um produto escalar é a multiplicação elementar entre o *patch* do tamanho do filtro da entrada e o filtro, que é então somado, sempre resultando em um único valor. Como isso, resulta em um único valor, a operação é geralmente chamada de produto escalar. Usar um filtro menor do que a entrada é intencional, pois permite que o mesmo filtro (conjunto de pesos), seja multiplicado pela matriz de entrada várias vezes em pontos diferentes na entrada. Especificamente, o filtro é aplicado sistematicamente a cada parte sobreposta ou *patch* do tamanho do filtro dos dados de entrada, da esquerda para a direita, de cima para baixo.

Esta aplicação sistemática do mesmo filtro através de uma imagem é uma ideia poderosa. Se o filtro for projetado para detectar um tipo específico de recurso na entrada, a aplicação desse filtro sistematicamente em toda a imagem de entrada permitirá ao filtro uma oportunidade de descobrir esse recurso em qualquer lugar da imagem. Esse recurso é comumente chamado de invariância de tradução, por exemplo, o interesse geral em saber se o recurso está presente, e não onde ele estava presente. A operação matemática que leva duas entradas, como matriz de imagem e como filtro ou *kernel*, é demonstrado pela a Figura 2.10.

A inovação das redes neurais convolucionais é a capacidade de aprender automaticamente um grande número de filtros em paralelo, específicos de um conjunto

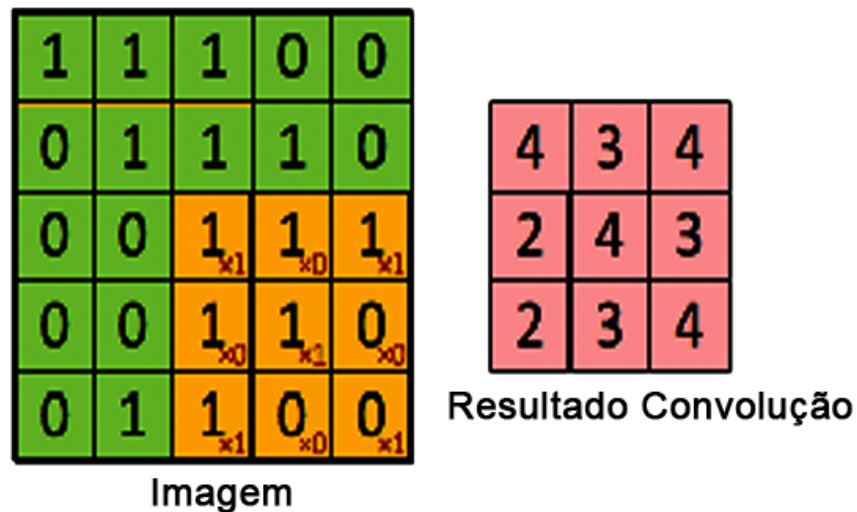


Figura 2.11 – Matriz de saída  $3 \times 3$ , resultante da convolução entre a matriz de imagem e do filtro ilustrado na figura 2.10.

de dados de treinamento, sob as restrições de um problema específico de modelagem preditiva, como a classificação de imagens. O resultado são recursos altamente específicos que podem ser detectados em qualquer lugar nas imagens de entrada (KALCHBRENNER; GREFFENSTETTE; BLUNSOM, 2014). Uma vez que um mapa de características é criado, podemos passar a ele cada valor através de uma não-linearidade, como uma ReLU, da mesma forma que fazemos para as saídas de uma camada totalmente conectada. Uma convolução e uma matriz de imagem de  $5 \times 5$  se multiplicada com uma matriz de filtro com  $3 \times 3$ , em que este procedimento é chamado de "Mapa de característica" gera um saída em que é demonstrada na Figura 2.11, a convolução de uma imagem com filtros diferentes pode realizar operações como detecção de bordas, desfoque e nitidez aplicando filtros.

### 2.3.2.2 Camada de combinação - (*pooling*)

As camadas convolucionais em uma rede neuronal convolucional aplicam sistematicamente filtros aprendidos a imagens de entrada para criar mapas de características que resumem a presença desses recursos na entrada. camadas convolucionais se mostram muito eficazes, e o empilhamento de camadas convolucionais em modelos profundos permite que camadas próximas à entrada aprendam recursos de baixo nível (por exemplo, linhas) e camadas mais profundas do modelo são usadas para aprender recursos de alta ordem ou mais abstratos, como formas ou objetos específicos. Uma limitação da saída do mapa de recursos das camadas convolucionais é que eles registram a posição precisa dos recursos na entrada. Isso significa que pequenos movimentos na posição do recurso na imagem de entrada resultarão em um mapa de recursos diferente. Isso pode acontecer com o recorte, rotação, deslocamento e outras pequenas alterações na imagem de entrada (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Uma abordagem mais robusta e comum é usar uma camada de combinação de

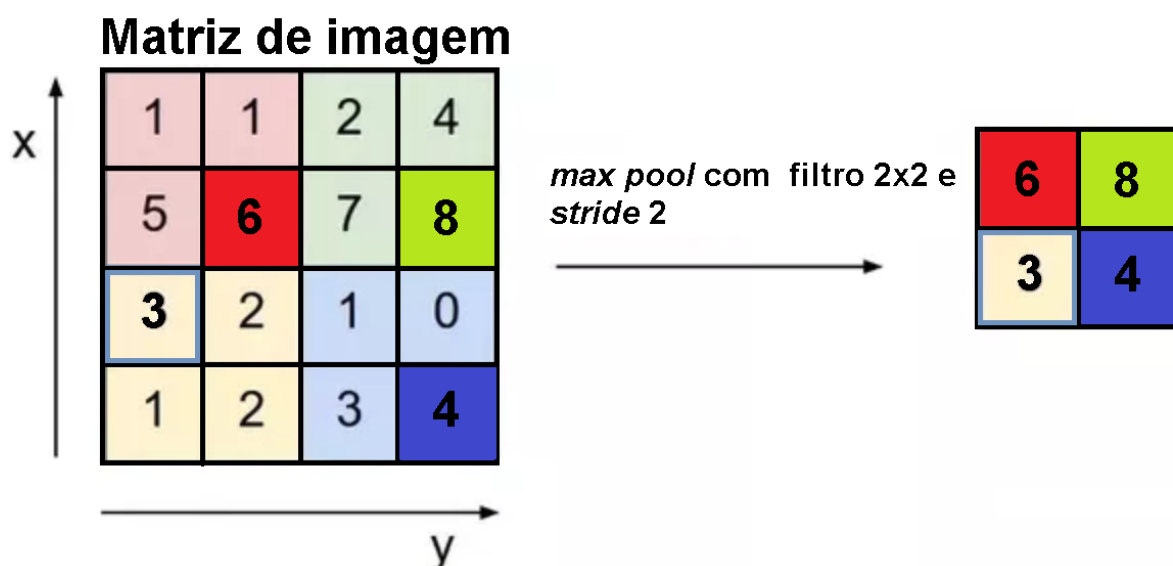


Figura 2.12 – Operação de *max pooling* entre uma matriz de imagem e um filtro  $2 \times 2$  com *stride 2*.

parâmetros (*pooling*), que é uma nova camada adicionada após a camada convolucional. Especificamente, depois de uma não-linearidade (por exemplo, ReLU), em que se aplica aos mapas de recursos emitidos por uma camada convolucional; por exemplo, as camadas de um modelo podem ter as seguintes simulações: Uma imagem de entrada, camada convolucional, não-linearidade e camada de combinação. A adição de uma camada de combinação após a camada convolucional é um padrão comum usado para ordenar camadas dentro de uma rede neuronal convolucional que pode ser repetida uma ou mais vezes em um dado modelo.

A camada de *pooling* processa cada mapa de recursos separadamente para criar um novo conjunto do mesmo número de mapas de recursos agrupados. A combinação envolve a seleção de parâmetros advindos de camadas anteriores, por meio de um filtro a ser aplicado aos mapas de recursos (HE et al., 2016). O tamanho da operação de combinação ou filtro é menor que o tamanho do mapa de recursos; especificamente, é quase sempre uma matriz de tamanho  $2 \times 2$  pixels aplicados com um passo de 2 pixels.

Isso significa que a camada de combinação *pooling* sempre reduzirá o tamanho de cada mapa de recursos por um fator de 2, por exemplo, cada dimensão é reduzida à metade, reduzindo o número de pixels ou valores em cada mapa de recursos para um quarto do tamanho. Por exemplo, uma camada de combinação aplicada a um mapa de recursos de  $6 \times 6$  (36 pixels) resultará em um mapa de recursos agrupados de saída de  $3 \times 3$  (9 pixels). Na Figura 2.12, uma matriz ilustrativa de imagem é filtrada por um filtro  $2 \times 2$ , onde a operação de *max pooling* seleciona os maiores valores da imagem.

A operação de combinação é especificada, em vez de aprendida. Duas funções comuns usadas na operação são: *pooling* médio : calcule o valor médio de cada *patch* no mapa de recursos. Combinação máxima (ou *max pooling*) : calcule o valor máximo para cada *patch* do mapa de recursos. O resultado do uso de uma camada de combinação e

da criação de mapas de recursos amostrados ou combinados é uma versão resumida dos recursos detectados na entrada. Eles são úteis, pois pequenas mudanças na localização do recurso na entrada detectada pela camada convolucional resultarão em um mapa de recursos combinados com o recurso no mesmo local. Esse recurso adicionado pela combinação é chamado de invariância do modelo para a tradução local.

### 2.3.2.3 Camada Totalmente Conectada

Uma camada totalmente conectada têm conexões completas com todas as ativações na camada anterior. Suas ativações podem, portanto, ser computadas com uma multiplicação de matrizes seguida por um deslocamento de polarização (HE et al., 2015). Essa camada basicamente recebe um volume de entrada (qualquer que seja a saída da convolução) e gera um vetor  $N$  dimensional onde  $N$  é o número de classes que o programa deve escolher. Por exemplo, se você quisesse um programa de classificação de dígitos,  $N$  seria 10, pois há 10 dígitos. Cada número neste vetor de dimensão  $N$  representa a probabilidade de uma determinada classe. Por exemplo, se o vetor resultante para um programa de classificação de dígitos para  $[0 \ .1 \ .1 \ .75 \ 0 \ 0 \ 0 \ 0 \ .05]$ , então isso representa uma probabilidade de 10% de que a imagem seja 1, uma probabilidade de 10% de que a imagem é um 2, uma probabilidade de 75% de que a imagem é um 3 e uma probabilidade de 5% de que a imagem seja um 8. A maneira como essa camada totalmente conectada funciona é que ela examina a saída da camada anterior (deve representar os mapas de ativação de recursos de alto nível) e determina quais recursos estão mais correlacionados a uma determinada classe.

Teoricamente, modelos de redes neuronais convolucionais de aprendizagem profunda em seu processo de treinamento e teste, cada imagem de entrada passará por uma série de camadas de convolução com filtros (*Kernels*), *Pooling* (agrupamentos), camadas totalmente conectadas e aplicação da função *Softmax*, em que para classificar um objeto com valores probabilísticos entre 0 e 1. A figura 2.13 ilustra o fluxo completo de uma RNC, em que processa uma imagem de entrada e classifica os objetos com base nos valores de entrada.

### 2.3.3 Redes Residuais

Aprendizagem residual profunda (HE et al., 2016) se desenvolveu da tentativa de se aumentar o número de camadas em uma rede convolucional. Para realizar a extensão de uma rede neuronal convolucional (RNC), pode-se por exemplo, adotar uma estratégia de passagem dos filtros de convolução. Em uma prática regularmente adotada, aplica-se, por exemplo, um filtro  $3 \times 3$  a cada 2 pixels da imagem (*stride 2*). Ao fim de tal operação, a extensão pode ser feita ao adicionar uma nova camada de convolução onde, ao invés de a cada 2 pixels, a passagem de filtro dá-se pixel a pixel (*stride 1*). Com tal estratégia de exemplo, é possível aumentar o número de camadas de uma rede RNC, onde cada camada

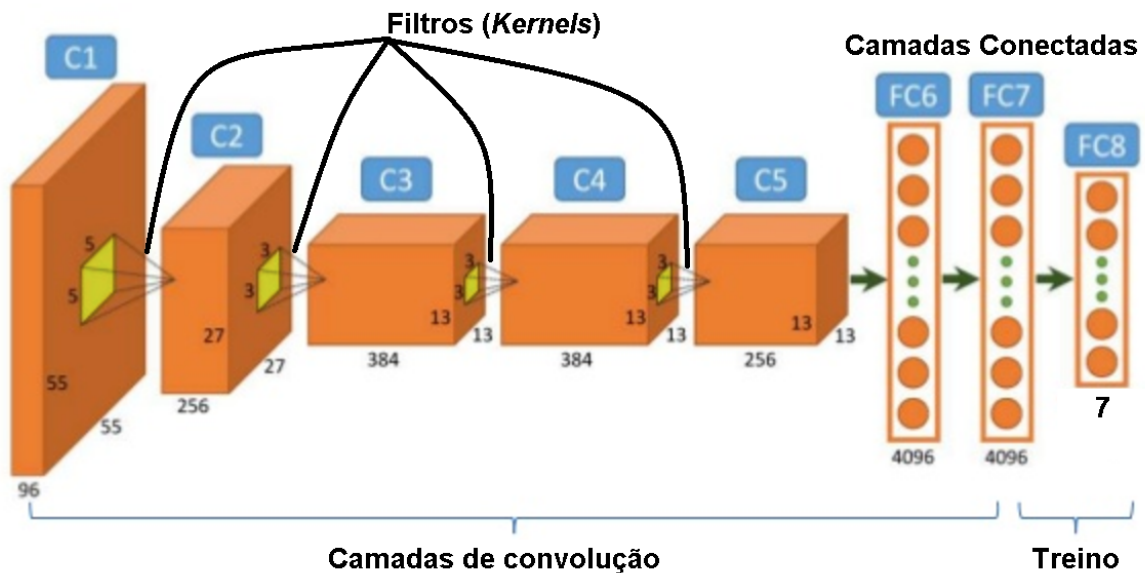


Figura 2.13 – Modelo de uma rede neural convolucional típica.

é conectada somente às camadas contíguas.

Entretanto, as expectativas de (HE et al., 2016) de se obter melhores resultados com extensão de redes RNC, no contexto de reconhecimento de imagens, foi frustrada em um primeiro momento. Esperava-se de uma rede com mais parâmetros um sobreajuste, de forma que em poucas épocas (abaixo de dois dígitos), se pudesse observar um erro de treinamento próximo a zero. Isto se deu, uma vez que as redes com mais camadas de convolução tendem a apresentar maior erro de treinamento do que uma rede significativamente menor (HE et al., 2016). Com mais camadas as redes tendem a produzir uma descida de gradiente praticamente nula em poucas épocas de treinamento, indicando uma alta susceptibilidade à degradação dos seu parâmetros.

A solução proposta, sugeriu que, a cada duas camadas de convolução, o sinal processado não seguisse por um único caminho, ou seja, o sinal de entrada seguia para a primeira camada e o resultado da operação era processado conseqüentemente, na camada contígua, resultando em um sinal de saída. Ao invés de tal operação puramente em cadeia, o sinal de entrada inicial alimenta a primeira camada, a qual realiza sua operação e alimenta a próxima camada, onde agora o seu sinal será combinado ao sinal de entrada anterior às duas camadas. Assim, operação anterior pode ser descrita como  $y = conv_1(conv_2(x))$ , com  $x$  sendo a entrada inicial e  $y$  a saída final processada através das camadas de convolução em cadeia. Já a solução proposta pode ser descrita como  $y = x + conv_1(conv_2(x))$ , onde  $y$  não mais é apenas o resultado em cadeia das camadas de convolução, e sim, uma combinação do resultado das camadas de convolução exemplificadas com a entrada inicial  $x$ .

A Figura 2.14a exemplifica uma operação em cadeia sem resíduo, onde o primeiro bloco é uma sequência em cadeia (convolução de filtro  $3 \times 3$ , *batch normalization* (SANTURKAR et al., 2018), ReLU (NAIR; HINTON, 2010)) seguida por duas convoluções. Já



na Figura 2.14b, pode-se observar igualmente nos primeiros blocos a propositura residual onde a entrada segue por dois caminhos: o primeiro segue através de uma convolução de filtro  $3 \times 3$ , *batch normalization*, ReLU e termina em uma camada de convolução de filtro  $1 \times 1$ ; o segundo caminho é o resultado da primeira camada de convolução de filtro  $3 \times 3$ . Assim o resultado dos dois caminhos é combinado por concatenação (podem ser realizadas outras operação como, adição por exemplo).

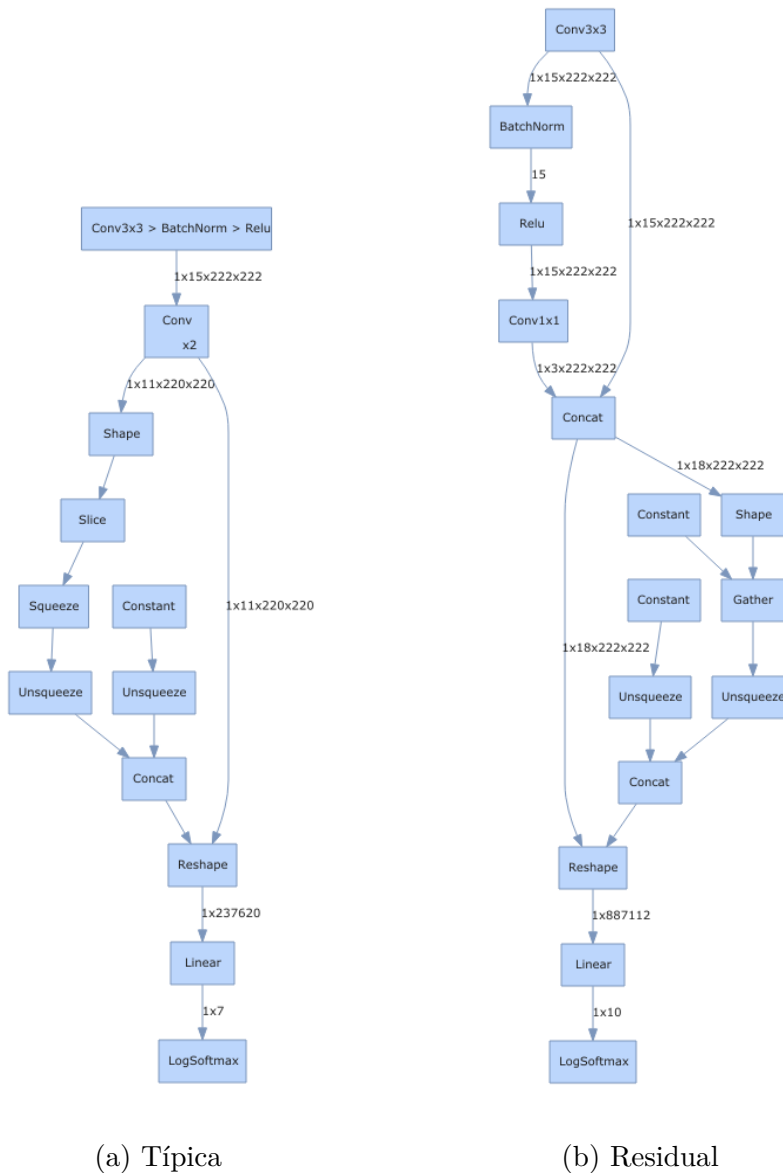


Figura 2.14 – Redes convolucionais, a) Típica, b) Residual.

Neste Capítulo, portanto, foram abordadas as principais iniciativas em reconhecimento de insetos relacionados a lavouras bem como as principais pragas do milho com suas características de ataque à lavoura. Ainda, destacou-se os principais trabalhos relacionados a identificação de insetos a partir de redes neuronais e as principais funções utilizadas para construção de arquiteturas de redes neuronais clássicas, convolucionais e residuais. No Capítulo 3 encontram-se os métodos utilizados para reunir e organizar imagens de pragas do milho, tanto para as consideradas pragas primárias quanto para as secundárias. Ainda

no Capítulo 3, encontram-se também os principais detalhes concernentes às arquiteturas de redes neuronais em cada uma de suas características.

## 3 Materiais e Métodos

O Capítulo 3 compreende: a construção do conjunto de imagens, a criação e desenvolvimento do sistema de reconhecimento de pragas baseado em redes neurais convolucionais, com uso da biblioteca Fastai (HOWARD, 2018) e a descrição das métricas utilizadas na avaliação dos resultados. A organização do novo conjunto de dados, bem como as adaptações em um modelo Inception-V3\*, em que foi comparado com redes residuais de alto desempenho, são contribuições deste trabalho.

### 3.1 Construção de um Conjunto de Imagens de pragas de milho em campo aberto

A escolha para a construção de cada conjunto de imagens de pragas seguiu a divisão proposta em (SCHNEIDER et al., 2015), pragas primárias e secundárias. Considerou-se também, a disponibilidade das pragas registradas em bancos localizados na *internet*. Além das imagens de pragas, adicionou-se uma categoria de imagens contendo apenas plantações, sem a presença de pragas. A inserção desta categoria é de grande relevância para se estabelecer uma contra-prova às imagens com pragas. Pois é necessário conferir se há distinção clara entre o reconhecimento de pragas e imagens com apenas plantações. Assim, pode-se dizer que fazer tal distinção, é um parâmetro comparativo, necessário para a confiabilidade dos resultados obtidos durante o treinamento das redes neurais.

#### 3.1.1 Aquisição de imagens de pragas

A aquisição de imagens se deu por meio de fotografias tomadas em campo aberto, em maioria disponibilizadas na *internet*. Como as imagens foram disponibilizadas por diferentes pesquisadores da entomologia em diversas partes do mundo e fornecidas como cortesia para o desenvolvimento do trabalho, foi necessário realizar busca e captura de imagens das pragas categorizadas na *internet* a partir do nome das espécies (VASA, 2018). Uma vez reunidas localmente e já separadas de acordo com a categoria, onde cada uma possui uma quantitativo inicial de 100 imagens, cada classe tem seu número de imagens balanceado para atingir um total 720 imagens. O balanceamento inicial se dá através

da aplicação de transformações geométricas e radiométricas (rotação ( $90^\circ$ ,  $-90^\circ$  e  $180^\circ$ ) e intensidade). Tal medida é considerada uma estratégia a se seguir quando há escassez de dados. Cada imagem resultante, possui alteração espacial dos pixels através rotação, enquanto a variação de intensidade de pixel produz efeito de variação de luz.

Em seguida, estabeleceu-se o formato PNG (*Portable Network Graphics*) e tamanho padrão  $224 \times 224 \times 3$  para todas as imagens. Finalizados os preparativos iniciais, dividiu-se o conjunto de imagens em dois subgrupos: pragas primárias e todas as pragas (primárias e secundárias). Em cada subgrupo, há um diretório destinado a cada praga, e um diretório extra com imagens de plantas, sem a presença de pragas do milho. Assim, a Tabela 2 representa a distribuição das imagens iniciais já divididas de acordo com as duas fases, treino e validação, totalizando 700 imagens (560 de treino, 140 de validação).

Tabela 2 – Conjunto inicial de imagens de pragas do milharal.

<b>Espécies</b>	<b>#Orig.Treino</b>	<b>#Orig.Validação</b>
<i>Agrotis ipsilon</i>	80	20
<i>Diatraea saccharalis</i>	80	20
<i>Helicoverpa zea</i>	80	20
<i>Spodoptera frugiperda</i>	80	20
<i>Dalbulus maidis</i>	80	20
<i>Diabrotica speciosa</i>	80	20
<i>Sem praga</i>	80	20
<b>TOTAL</b>	<b>560</b>	<b>140</b>

### 3.1.2 Pré-processamento para ampliação do conjunto de Imagens

Em posse do conjunto de imagens iniciais, deu-se prosseguimento à fase de ampliação do conjunto neste processo, os subconjuntos de 100 imagens de cada pragas é ampliado até chegar a um total de 720 amostras. Cada uma das 100 imagens pertencentes às espécies dá origem a pelo menos 7 a 8 novas imagens.

Foram aplicadas 4 grupos de transformações, onde cada uma das transformações produziu pelo menos duas imagens. A primeira transformação agrupa em si uma pequena série de transformações: rotação normal e aleatória de  $10^\circ$  combinada com luminosidade, também aleatoriamente variável em até 5%. Ao resultado de tal combinação, acrescentou-se uma rotação aleatória de até  $15^\circ$  ao longo do eixo  $y$  da imagem. Finalmente, esta primeira transformação termina com a adição de uma *zoom* variável de até 10%. A segunda transformação aplicou até 15% de *zoom* sobre cada uma das imagens iniciais. Em seguida, a terceira transformação produziu sua parcela de amostras com variações de iluminação de até 50%. E finalmente, a quarta transformação tenciona (efeito de objeto esticado)

cada imagem em proporções aleatórias de até 15%. Desta forma, a Tabela 3 apresenta a quantidade para treino e validação nas pragas do conjunto final já ampliado. O total de 4320 imagens inclui apenas as pragas e, ao considerar-se a categoria de imagens sem pragas, tem-se 5040 imagens utilizadas em cada um dos dois grupos de experimento: primárias e todas (primárias e secundárias além das imagens sem pragas).

Tabela 3 – Total de imagens no conjunto final após as transformações para ampliação.

Identificador	Espécies	# Treino Ampliado	# Valid. Ampliada
1	<i>Agrotis ipsilon</i>	560	160
2	<i>Diatraea saccharalis</i>	560	160
3	<i>Helicoverpa zea</i>	560	160
4	<i>Spodoptera frugiperda</i>	560	160
5	<i>Dalbulus maidis</i>	560	160
6	<i>Diabrotica speciosa</i>	560	160
7	Sem praga	560	160
TOTAL 3920	1120		

A figura 3.1 demonstra os procedimentos realizados no conjunto de dados de imagens até que se atingisse o balanceamento entre as classes de pragas, tanto para o conjunto de treinamento e validação. A harmonização numérica entre as classes de pragas foi obtido ao utilizar a técnica de ampliação *augmentation* descritas neste Capítulo: empregando rotação de 90°, rotação de -90°, rotação de 180°, zoom aleatório de até 1,5x, alongamento aleatório, rotação aleatória máxima dentro de um intervalo de 10° e, variação de intensidade de pixel para simular diferentes condições de iluminação.

Neste trabalho, utilizou-se a técnica de transferência de aprendizado para suprir a carência de dados de treino, além das técnicas de ampliação já mencionadas na Seção 3.1. A Figura 3.2 sintetiza os principais passos em aprendizagem supervisionada.

## 3.2 Arquiteturas de Redes Neurais

Disponível o banco de imagens para treinamento, é possível configurar os parâmetros da redes neurais escolhidas para o processo de aprendizagem. As redes podem ser divididas em três categorias: clássicas, convolucionais típicas e convolucionais residuais.

### 3.2.1 Modelo de Rede Neuronal Clássica - Máquina de Vetores de Suporte SVM-LBP

Uma arquitetura de rede do tipo máquina de vetores de suporte (SVM) é utilizada com frequência em problemas de classificação de imagens na agricultura (EBRAHIMI et al., 2017). Para se obter uma SVM, é possível modificar a função de minimização de erro em uma rede neuronal. Dada uma função de minimização de erro em regressão

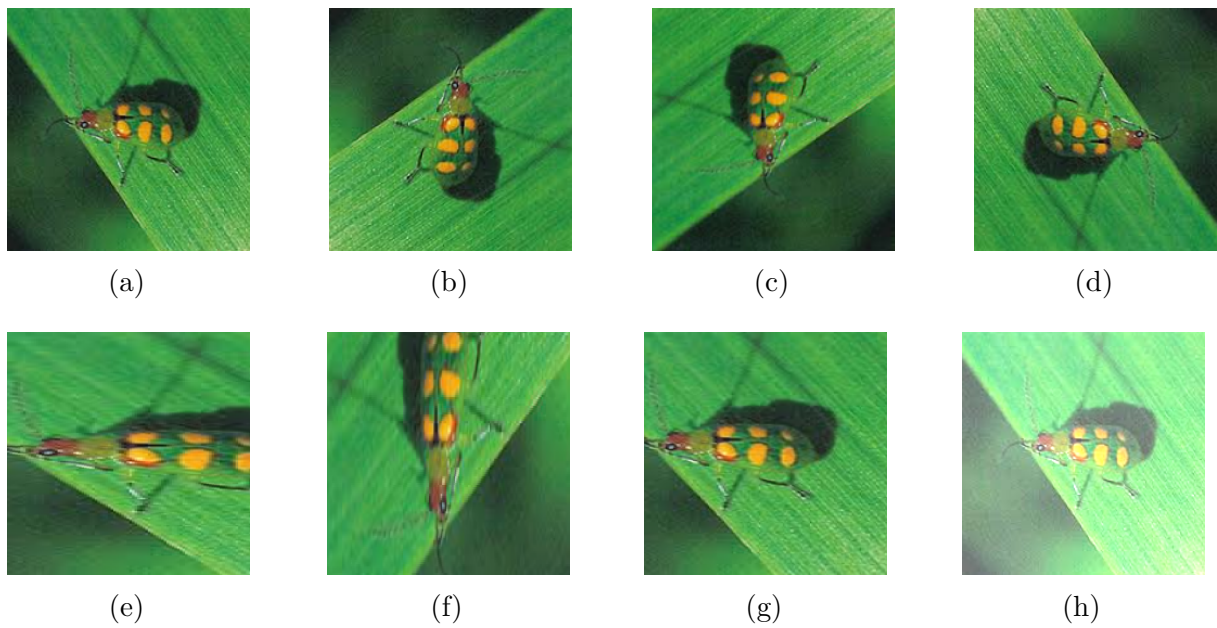


Figura 3.1 – Processos de transformações realizadas em imagens do conjunto de dados, para obter o balanceamento entre as classes. (a) Imagem original, (b) Rotação de 90°, (c) Rotação de -90°, (d) Rotação de 180°, (e) Zoom aleatório de até 1,5x, (f) Alongamento aleatório, (g) Rotação aleatória máxima dentro de um intervalo de 10° e (h) Variação de intensidade de pixel para simular diferentes condições de iluminação.

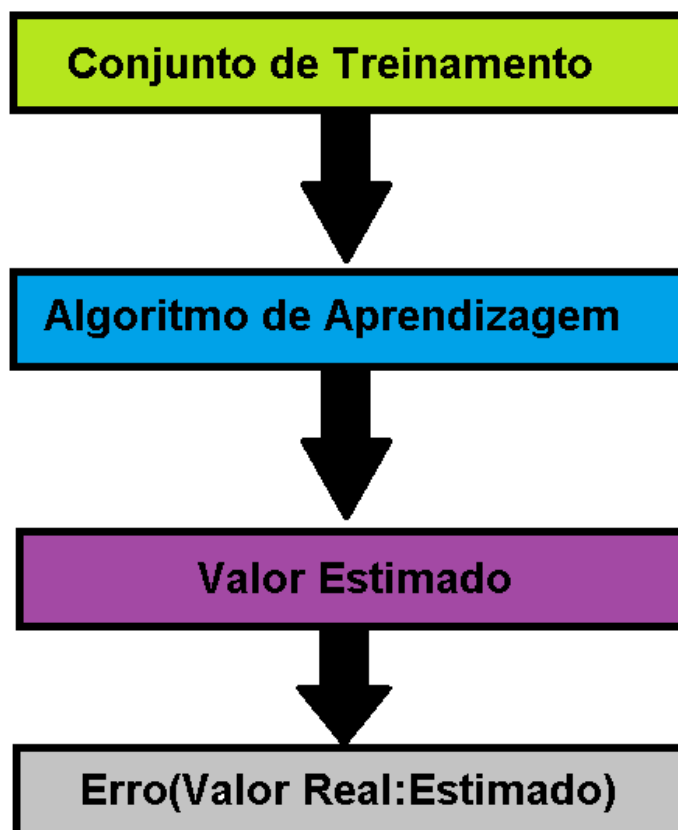


Figura 3.2 – Principais passos para realização de aprendizagem supervisionada.

$J(\theta) = \min(A + \lambda B)$  onde pode-se restringi-las a classes separáveis como 0 ou 1, com A, a Equação 3.1 construída para aproximadamente resultar em 0 ou 1. Adicionalmente a B, define-se como uma termo de regularização, expresso na Equação 3.2. Em ambas as Equações 3.1 e 3.2,  $m$  refere-se à quantidade de amostras de treino,  $y$ , a variável dependente a ser predita,  $z = \theta^T X$  e  $\lambda$  um parâmetro de regularização para controle da relação de peso entre A e B. Desta forma, se  $\lambda$  possui um valor muito expressivo, então B passa a ser tão ou mais relevante que A, já quando o contrário é aplicável, então A tem maior relevância que B para um possível valor ótimo para  $\theta$ .

$$A = \sum_{i=1}^m -y_i \log\left(\frac{1}{1 + e^{-z}}\right) + (1 - y_i) \log\left(\frac{1}{1 + e^{-z}}\right) \quad (3.1)$$

$$B = \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \quad (3.2)$$

Ainda, modifica-se também para a SVM, a função de minimização para que, por convenção, esta seja  $J(\theta) = \min(CA + B)$  sem prejudicar seu potencial de conduzir a convergência ao valor ótimo para  $\theta$ . Assim, C tem função inversa a  $\lambda$ . Por conseguinte, a função preditiva de uma SVM produzirá valor para igual a 1, se  $\theta^T X \geq 0$  ou 0, caso contrário. Dadas as condições de definição de uma SVM, esta pode ser ilustrada na Figura 3.3, onde a camada de entrada pode conter uma representação de  $X$  (conjunto de dados de entrada), de forma que este possa ser separável, seja por linearidade ou por funções não-lineares. Por conseguinte, pode-se estender a função nuclear para que a SVM seja aplicável para várias classes dentro de um conjunto de treino. Uma possibilidade para aplicar separação de imagens de acordo com as características dos valores dos pixels das imagens é binarização por localidade.

O LBP produz um histograma dos canais de uma imagem, mapeando cada pixel em zero quando a intensidade do pixel é abaixo de 128 ou um, caso esteja acima daquele valor. Tem-se, portanto, um raio de aplicação, onde os vizinhos pertencentes a este raio são mapeados em sentido anti-horário, formando uma sequência binária, como observado no exemplo proposto na Figura 3.4 em que um canal de uma imagem tem certa região mapeada para uma sequência binária  $0011100001110101_2$  ou  $14453_{10}$ . Por ser aplicável somente em um canal, uma vez alimentada com as imagens do problema proposto neste trabalho, estas foram convertidas para escala de cinza para que fossem extraídos os padrões binários em cada imagem. Desta maneira, a matriz em escala de cinza (à esquerda) representa uma seção de uma imagem, ao centro, tem-se sua representação numérica nas respectivas posições de cada cor. Por fim, a matriz à direita na Figura 3.4 apresenta a operação de de binarização efetuada sobre a imagem, onde valores em um determinada vizinhança acima de 127 são mapeados para o valor 1 e valores abaixo para o valor 0, de forma que se forme uma sequência binária a ser lida no sentido anti-horário.

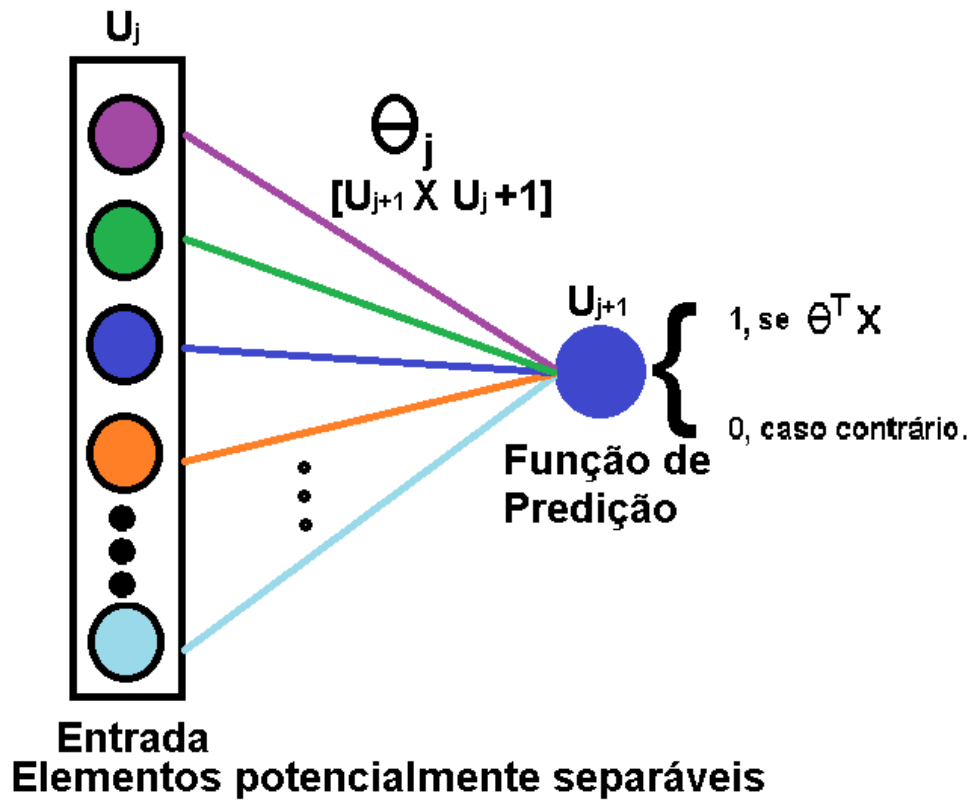


Figura 3.3 – Estrutura básica de uma SVM. Uma camada de entrada com elementos potencialmente separáveis em um hiperplano; um conjunto de parâmetros e uma função de hipótese.

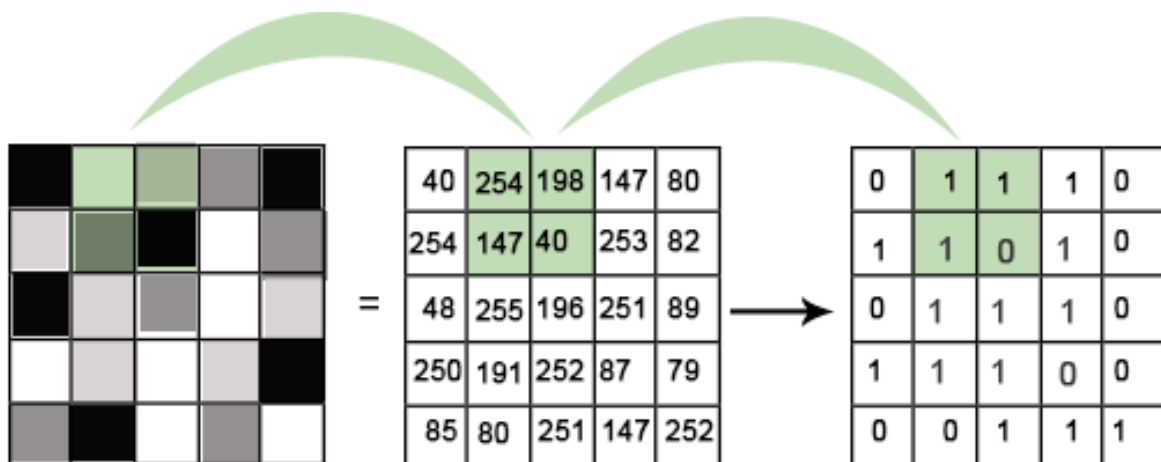


Figura 3.4 – Extração de características de padrão binário (LBP) em um canal utilizada com extração de características no modelo SVM.



Assim, é possível utilizar as características extraídas pelo LBP como entrada no modelo SVM. Isso faz com que não seja necessário utilizar diretamente as imagens, as quais por natureza, têm características complexas. O LBP é uma das possibilidades de técnicas de extração de informações estruturadas presente nas imagens.

### 3.2.2 Modelo de Rede Convolutiva Típica - AlexNet

O modelo AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) é uma variação de redes neurais convolucionais. Seu propósito está inserido dentro do contexto de paralelização de modelos de rede e de conjuntos de dados (KRIZHEVSKY, 2014). Quando se escolhe a paralelização do modelo, preocupa-se principalmente com a sincronização das saídas e processadas em cada núcleo, já quando ocorre a paralelização do conjunto de dados, a atenção se concentra na sincronização do valor dos parâmetros em cada núcleo, a fim de garantir consistência de treinamento (KRIZHEVSKY, 2014).

Em geral, redes convolucionais são predominantemente compostas por duas camadas principais: convoluções e camadas totalmente conectadas (nesta última, todas as saídas da camada anterior se conectam a cada uma das entradas da camada contígua subsequente), (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Camadas de convolução compõem a maior parte de uma rede, cerca de 95%, entretanto, estas são responsáveis por apenas 5% do total de parâmetros da rede. Por outro lado, camadas totalmente conectadas representam 5% de rede, porém atendem por cerca de 95% dos parâmetros presentes nesta, (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). De posse destas proporções, o modelo AlexNet visa abrigar tanto a paralelização dos dados quanto a paralelização do conjunto de dados. A paralelização do conjunto de dados especificamente se dá através das camadas de convolução, por serem majoritárias e possuírem a menor quantidade de parâmetros, são distribuídas através dos núcleos, onde cada convolução encontra-se operando sobre um *mini-batch* diferente (assíncrono). Já a paralelização do modelo é feita por meio das camadas totalmente conectadas, de forma que todas as camadas conectadas encontram-se no mesmo *mini-batch*, porém distribuídas ao longo dos núcleos, os quais devem manter-se sincronizados.

Assim, as etapas de propagação e retro-propagação possuirão estágios de intermitência durante o treinamento. Em dada época de exemplificação, a propagação do sinal é feita inicialmente por meio da paralelização dos dados nas camadas de convolução distribuídas através dos núcleos. Assim, os sinais propagados em um conjunto de camadas de um núcleo alimentam as diversas partes das camadas conectadas as quais estão distribuídas através dos núcleos. Uma vez computadas as ativações das camadas conectadas, é realizada a retro-propagação do sinal obtido, agora em direção às camadas de convolução.

O modelo AlexNet possui oito camadas, as cinco primeiras são convolucionais e as três últimas, camadas totalmente conectadas, com um total de 60 milhões de parâmetros distribuídos ao longo da arquitetura, (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

A última camada possui uma função *softmax*, responsável por gerar a distribuição das probabilidades para as 1000 classes do problema originalmente proposto. Entretanto, para o contexto do presente problema de reconhecimento de pragas do milho, a última camada com as 1000 saídas originais é substituída, hora com cinco saídas na primeira fase do experimento (quatro pragas primárias e uma sem pragas), hora com sete saídas, para atender a segunda fase do experimento, onde são testadas as seis pragas juntamente com as imagens contendo apenas plantas. A Figura 3.5 apresenta a arquitetura do modelo AlexNet devidamente adaptado ao presente contexto, onde a última camada foi redimensionada para o experimento com 7 categorias.

### 3.2.3 Modelos de Redes Convolucionais Residuais

As primeiras proposições, apesar de obter resultados expressivos, como a AlexNet, requer maior poder computacional, principalmente memória, devido aos seus 60 milhões de parâmetros. Devido ao alto custo computacional, foram propostos modelos de propósito geral, de considerável eficiência e quantidade muito menor de parâmetros, como o modelo Inception do GoogleLeNet (SZEGEDY et al., 2015), o qual é 12 vezes menor que o modelo AlexNet, com 5 milhões de parâmetros.

Entretanto, a eficiência do modelo é considerada difícil de ser reproduzida para outros estudos de caso, pois as regras utilizadas no projeto do modelo Inception não foram clarificadas (SZEGEDY et al., 2016). Portanto, as mudanças necessárias para aumentar a quantidade de camadas resulta desta forma, em aumento do custo computacional de modo significativo. Por exemplo, dobra-se a quantidade de camadas e logo tem-se o quádruplo de custo computacional e de parâmetros no modelo.

#### 3.2.3.1 Inception-V3

O Inception-V3 foi proposto sob princípios bem definidos (SZEGEDY et al., 2016) de uso parcimonioso. Primeiramente, a informação deve fluir desde a entrada da rede até sua saída sem encontrar gargalos, compressão agressiva dos dados, pois a compressão não garante que características relevantes da informação sejam corretamente preservadas para o uso das redes (SZEGEDY et al., 2016). Por outro lado, representações da informação em alta dimensionalidade pode permitir que redes convolucionais sejam treinadas mais rapidamente (SZEGEDY et al., 2016). Em seguida, dados de baixa dimensionalidade podem ser agregados, sem prejuízo à eficiência do modelo, devido à possível correlação dos dados em suas respectivas vizinhanças (SZEGEDY et al., 2016). E finalmente, deve haver um equilíbrio proporcional entre a largura e a profundidade de uma rede para se obter melhor eficiência das redes (SZEGEDY et al., 2016).

No modelo Inception original podem ser observadas convoluções de dimensão  $5 \times 5$  as quais comparadas às de dimensão  $3 \times 3$ , têm custo  $\frac{25}{9} = 2,77$  maior. Assim, ao se

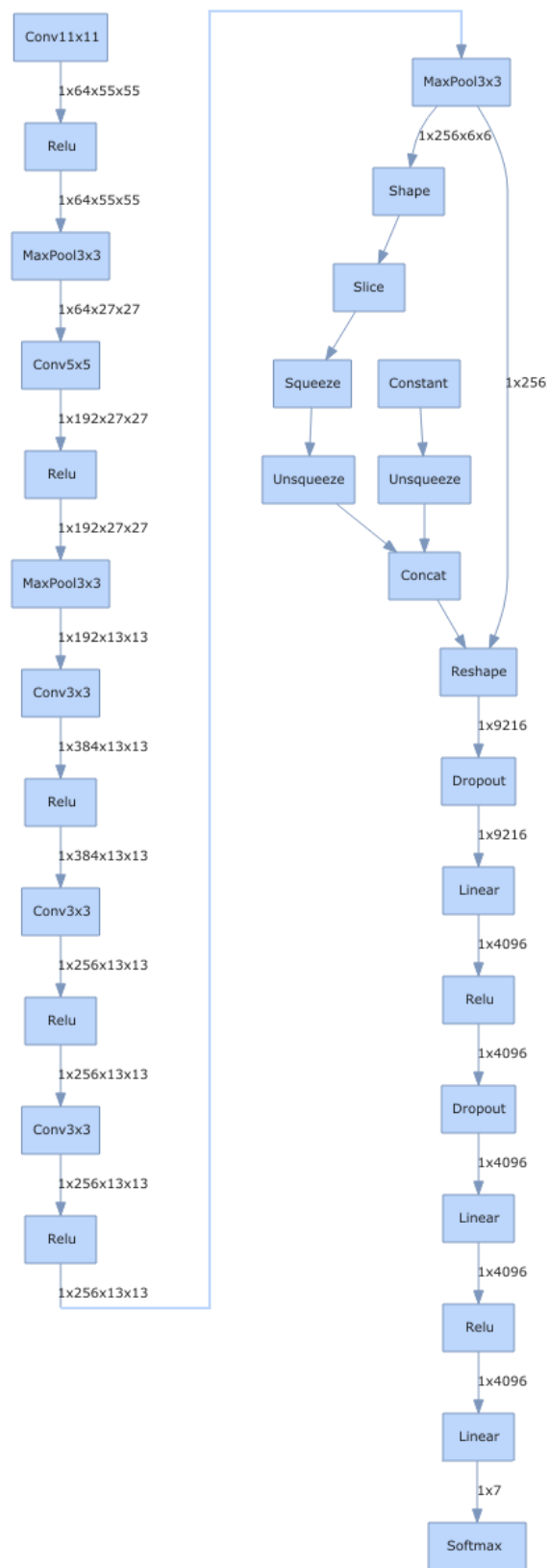


Figura 3.5 – Modelo AlexNet adaptado com 7 saídas probabilísticas através da função *softmax* (4 pragas primárias, 2 secundárias, 1 sem pragas.)

considerar o princípio de agregação de baixa dimensionalidade, pode-se substituir uma convolução de maior custo, como a de  $5 \times 5$ , por duas convoluções em cadeia de  $3 \times 3$ , obtendo resultado sem prejuízo à informação e com redução em custo de processamento. Desta forma, camadas de maior dimensão  $5 \times 5$  para a variação Inception-V3 pode ser substituídas por mini redes neuronais (duas camadas de convolução  $3 \times 3$ ) para redução de custo computacional em  $\frac{9+9}{25} \times 100\% = 28\%$  (SZEGEDY et al., 2016).

A substituição de camadas de convolução de maior dimensão por mini redes de camadas com menor dimensão é utilizada nas primeiras camadas do modelo Inception-V3 (SZEGEDY et al., 2016). Para as camadas intermediárias, tal abordagem foi repensada, de modo que suas camadas obtiveram maior resultado com camadas assimétricas do tipo  $1 \times n$  e  $n \times 1$ , por exemplo,  $1 \times 7$  seguida por uma convolução  $7 \times 1$ , (SZEGEDY et al., 2016).

Ainda, no modelo Inception (SZEGEDY et al., 2015), introduziu-se o conceito de dois classificadores ao fim da rede (dois ramos de saídas distintas) para contribuir para convergência características iniciais em redes muito profundas (SZEGEDY et al., 2015). Entretanto, há uma divergência de entendimento quanto isso (SZEGEDY et al., 2016), pois outro entendimento defende que a remoção do classificador auxiliar, não compromete o desempenho final da rede. Antes, o possível impacto do classificador auxiliar aplica melhor regularização à rede, devido a sua função de descarte de ativações (*dropout*) (SZEGEDY et al., 2016).

Por fim, os gargalos no modelo Inception-V3 são evitados ao se substituir operações de agrupamento (*max pooling*- agrupamento por máximos ou *average pooling*-agrupamento por média) por um grupo constituído de: i) agrupamento e duas convoluções  $1 \times 1$  em paralelo na camada de entrada da operação, ii) após as duas convoluções de  $1 \times 1$ , seguem duas convoluções  $3 \times 3$ , uma com deslocamento pixel a pixel (*stride 1*) e a segunda com deslocamento a cada dois pixels (*stride 2*), iii) à camada de convolução de *stride 1* em ii), segue-se uma camada de convolução  $3 \times 3$  de *stride 2*. Assim, o resultado obtido em iii) é concatenado com a saída da convolução  $3 \times 3$  de *stride 2* em ii), juntamente com o agrupamento efetuado em i). Os três estágios descritos, por conseguinte, são a proposta para se evitar gargalos ou perdas de informações relevantes ao longo da rede, quando há necessidade de se aplicar operações de agrupamento (SZEGEDY et al., 2016).

Assim, a Figura 3.6 apresenta o gráfico da rede Inception-V3 de forma compactada com fim de se obter compreensão de sua macroestrutura. Um dos seus blocos de camadas em especial possui uma ordem a qual pode ser observada ao logo da rede. Este bloco, denominado pela abreviação **ConvBnReLU** se repete três vezes, como observado no canto inferior do bloco ( $\times 3$ ), onde cada bloco é composto por uma convolução, uma operação de normalização (*batch normalization*) seguida por uma transformação efetuada por uma função *ReLU*. O bloco **ConvBnReLU** está presente em diferentes partes da rede, sendo repetido em até três vezes, Figura 3.6.

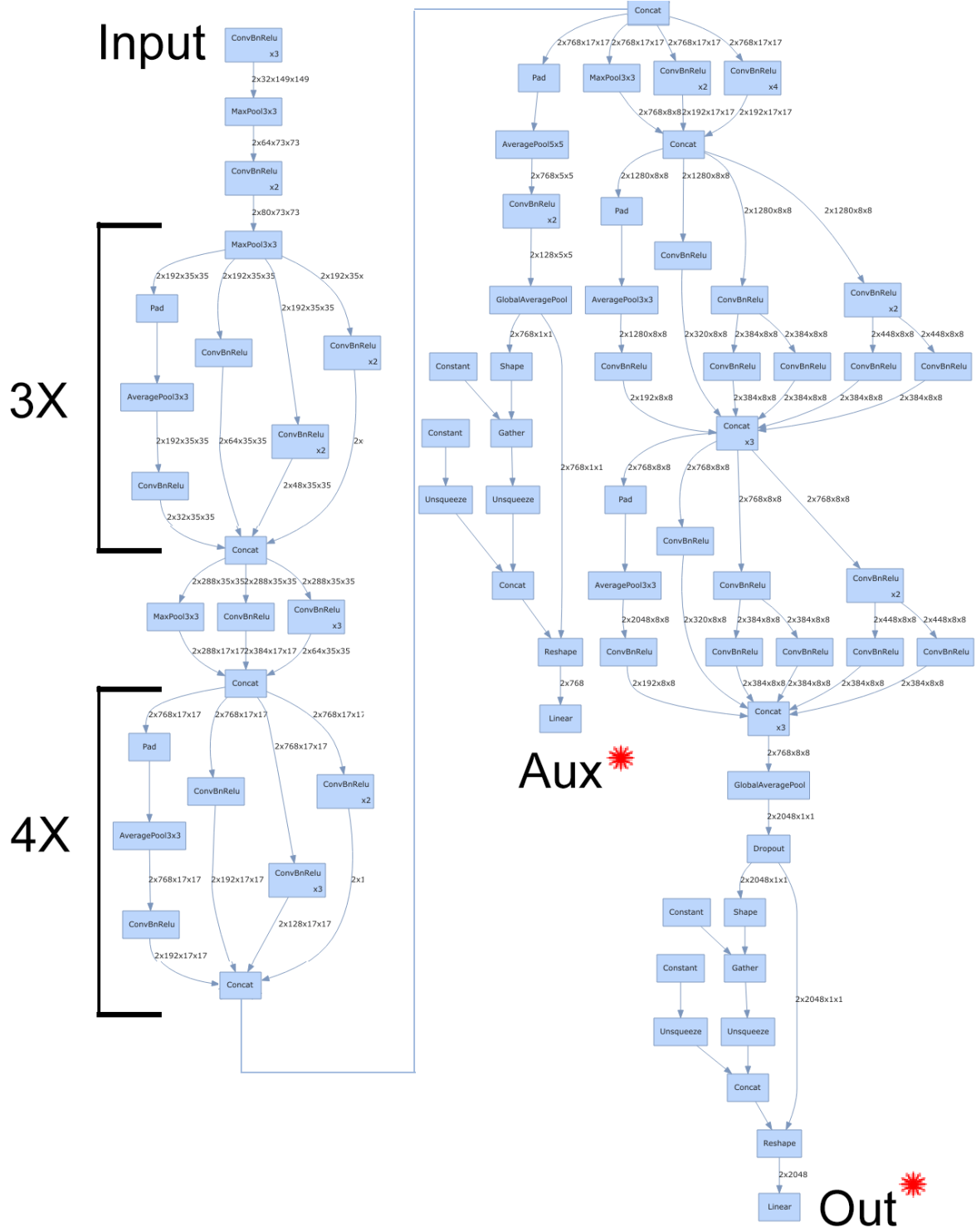


Figura 3.6 – Modelo Inception-V3 adaptado de (SZEGEDY et al., 2016) e compactado para melhor compreensão de sua macroestrutura. As dois marcadores (Aux\* e Out\*) remetem às modificações propostas e apresentadas em 3.2.3.2, concernentes aos dois classificadores nas saídas do modelo Inception-V3\*. A nova saída auxiliar (Aux\*) e a principal (Out\*) tiveram suas camadas Lineares substituídas por mini redes neurais.

### 3.2.3.2 Proposição a partir de um Modelo Residual Inception-V3\*

As modificações propostas neste trabalho ao modelo (Inception-V3) (SZEGEDY et al., 2015) visam estender sua arquitetura residual, a fim de se obter melhor desempenho no conjunto proposto de imagens de pragas do milho. O modelo original Inception-V3 possui um ramo de saída com classificador auxiliar opcional como apresentado na Figura 3.6. O modelo proposto Inception-V3\* foi modificado em suas duas saídas de classificadores com as seguintes estruturas de mini redes neuronais assim apresentadas na Figura 3.7: 1) camadas de regularização (*batch normalization* e descarte *dropout*) (SRIVASTAVA et al., 2014), 2) uma camada linear seguida por uma função de ativação (*ReLU*), 3) duas camadas de regularização (*batch normalization* e *dropout*), 4) uma camada linear a qual antecede ao classificador de saída da rede.

A razão de escolha das camadas bem como a sua ordem são como se segue: camadas de regularização proporcionam ao modelo um melhor aproveitamento da vantagem de se herdar os parâmetros pré-treinados do *Imagenet* para o novo caso de uso, sem abdicar da descida de gradiente efetiva durante a retropropagação da rede em cada mini-lote *mini-batch*. Ademais, as camadas de regularização com *dropout* substituem um percentual de parâmetros advindos de camadas anteriores com zeros. Esta substituição é realizada para se evitar que o modelo favoreça determinado conjunto de parâmetros ao formar rotas rotineiramente recorrentes ou "viciadas" durante o treinamento; especialmente durante a redução mais delicada do gradiente de erro nas últimas épocas de treinamento, quando a rede está próxima de alcançar sua maior eficiência possível. Finalmente, a camada linear, situada entre as duas sequências de camadas de regularização, tem o fim de reduzir o número de ativações de 4096 para 2048, como visto na Figura 3.7 na saída principal (**Out\***). Tal redução busca seguir o princípio de combinação de informações sem causar maiores perdas de informações relevantes ao melhor desempenho da rede.

Pode-se observar, entretanto, na Figura 3.7, a ausência da camada linear intermediária descrita acima para a saída auxiliar (**Aux\***). Isto se deve ao fato de a camada auxiliar possuir apenas 768 parâmetros (18,75%) quando comprada à saída principal **Out\***. Por esta razão, manteve-se constante o número de parâmetros até a última camada linear a qual realiza a combinação dos parâmetros para atender a quantidade de classes do conjunto de imagens (5 na fase com pragas primárias e imagens sem pragas e 7 na fase com todas as pragas juntamente com as imagens sem pragas).

Embora as mini redes substituintes às saídas originais tenham estrutura semelhante (exceto pela ausência da camada linear intermediária em **Aux\*** devido ao já baixo número de parâmetro advindo de partes anteriores da rede), as camadas de *dropout* são aplicadas como valores distintos em cada saída. Especificamente para o caso de uso do conjunto de imagens de pragas do milho, as duas camadas de *dropout* na saída principal foram configuradas para descartar 20% e 10% respectivamente das ativações advindas de camadas anteriores da rede; na saída auxiliar, o descarte de ativações realizado nas camadas de

*dropout* foram de 30% e 15% respectivamente. Em ambas as saídas, os valores de descarte estabelecidos para a entrada principal (20% e 10%) e camada auxiliar (30% e 15%) foram escolhidos a partir de testes com combinações de valores entre 5% e 95% de descarte.

O descarte de parâmetros e ativações proporciona ao modelo maior generalização quanto a suas predições nos classificadores principal e auxiliar, quando aplicado ao novo conjunto de imagens proposto. Além disso, o descarte ajuda a reduzir a possibilidade de ocorrência de sobreajuste e degradação durante o treinamento, conduzindo a rede a um maior rendimento final. Portanto, as últimas camadas lineares originais (4096x1000 entrada/saída com operação de produto escala de matrizes da saída principal, ilustrado na Figura 3.6) e na saída auxiliar (768X1000 entrada/saída com 1000 classes) foi substituída por mini redes com classificadores devidamente dimensionados, Figura 3.7 para o conjunto de imagens propostos.

Este Capítulo descreveu portanto, os passos necessários para a construção do conjunto de imagens de campo concernentes às pragas primárias e secundárias coletadas. Ainda, apresentou -se também as técnicas de ampliação do conjunto, as quais elevaram o número inicial de 700 imagens para o total de 5040, já inclusas as imagens de plantas sem pragas, as quais são necessárias para tentar garantir a distinção entre primeiro plano e plano de fundo na tarefa de reconhecimento. Compõe adicionalmente este Capítulo uma breve descrição das arquiteturas de redes neuronais utilizadas e seus principais elementos como funções internas (convoluções, ativação, combinação, regularização e etc.), além das funções de otimização. No Capítulo 4, encontram-se os detalhes de cada fase de aplicação da metodologia desenvolvida para o reconhecimento das pragas estudadas, composta por um conjunto de parâmetros iniciais de treinamento e os ajustes de hiper-parâmetros em cada rede. Por fim, apresentam-se as tabelas referentes à exatidão obtida nas duas fases do experimento: reconhecimento de pragas primárias e imagens sem pragas e reconhecimento de todas as pragas juntamente com imagens sem pragas.

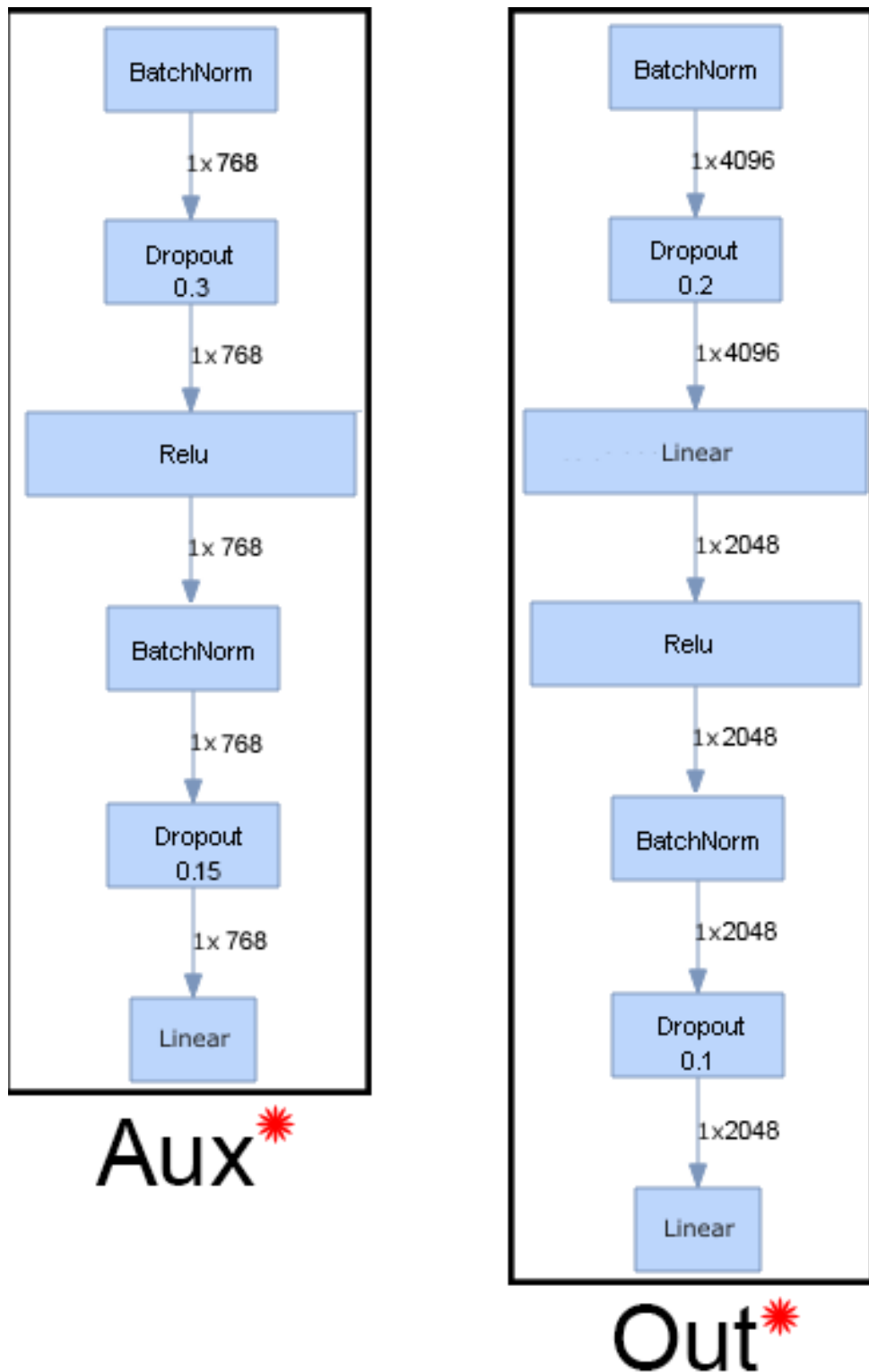


Figura 3.7 – Novas saídas (Aux\* e Out\*) do modelo proposto Inception-V3\* com camadas de regularização (*batch normalization* e *Dropout*), redução de parâmetro e ativações (Linear) camadas de ativação(*ReLU*).



## 4 Experimentos

O Capítulo 4 descreve os detalhes pertencentes às duas principais fases no problema de reconhecimento das pragas do milho: i) treinamento das redes no conjunto de pragas primárias e de imagens sem pragas, ii) treinamento das redes com todas as pragas selecionadas (primárias e secundárias) juntamente com as imagens sem pragas.

Durante cada uma das fases de treinamento, além da ampliação do conjunto de imagens descrita no Capítulo 3, aplicou-se também os recursos de transformação como variação de cor, tamanho, rotação, aproximação e iluminação nas disponíveis pelos algoritmos das redes, em transições de uma época para outra, nas imagens de entrada. ambos os experimentos foram conduzidos em uma placa gráfica GPU-*graphical process unit*(unidade de processamento gráfico) NVIDIA GeForce 1060, 16GB (dezesesseis *GigaBytes*) de *RAM* (*Random Access Memory -memória volátil*), com o auxílio da biblioteca de aprendizagem Fastai 0.7 (HOWARD, 2018) sobre um arcabouço Pytorch desenvolvido em linguagem *Python*.

### 4.1 Uso do conjunto de imagens durante o experimento

A biblioteca Fastai (HOWARD, 2018) permite que os modelos implementados possam gerar conjuntos de imagens a mais no conjunto de validação, por meio de transformações aleatórias. Isto permite explorar variações de iluminação, ângulo e translação que não estão presentes no conjunto.

Na primeira parte principal do experimento, com imagens de tamanho fixo de  $224 \times 224 \times 3$ , as cinco categorias (quatro pragas primárias e uma sem pragas) foram utilizadas no treinamento de todas as imagens. Na segunda parte do experimento, todas as espécies de pragas selecionadas foram testadas juntamente com as as imagens sem pragas.

Uma variação de ampliação virtual para o conjunto de validação possível foi testada devido às redes terem a possibilidade de realizar subamostragem de  $224 \times 224$  para  $112 \times 112$ , a fim de explorar o potencial das redes ao serem treinadas inicialmente com formas simples antes de detalhes mais complexos em imagens maiores. Formas geométricas mais simples são extraídas nas primeiras camadas de convolução das redes. Assim, como os conjunto de imagens, quando aplicadas ao SVM, são convertidas em escala de cinza, decidiu-

se explorar também tal condição nas RCNs como ilustrado na Figura 4.1a, onde aplicou-se uma combinação de subamostragem escala de cinza utilizada nas quatro primeiras épocas. A escala de cinza é uma transformação por média ponderada, combinando os três canais de uma imagem para o padrão de cinza NTSC descrito na Equação 4.1. Esta segunda transformação visa explorar a habilidade de cada modelo manter eficiência na classificação, quanto à forma e intensidade de cada pixel nas imagens, antes de se explorar variação de cor e de textura proveniente da cor.

$$X_{R,G,B} = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B, \quad \text{with,} \quad R = 0, \quad G = 1, \quad B = 2 \quad (4.1)$$

Onde  $X_{R,G,B}$  é a nova imagem com um canal de pixels de valor de média ponderada dos três canais originais em cada posição  $m \times n$  com  $m, n = 1, 2, 3, \dots, 224$  para o modelo AlexNet e até 299 para os modelos Inception-V3. No entanto, a variação de tamanho mostrou-se impraticável até o momento, devido à grande redução de precisão do tamanho regular 224 para o 112. Assim descartou-se também a combinação com as imagens com escala de cinza, devido ao maior número de épocas necessárias para o treinamento. Assim, mantiveram-se fixos os parâmetros de tamanho, 224 para SVM-LBP e AlexNet e 299 para Inception-V3 e Inception-V3\* por ser obrigatório para a arquitetura.

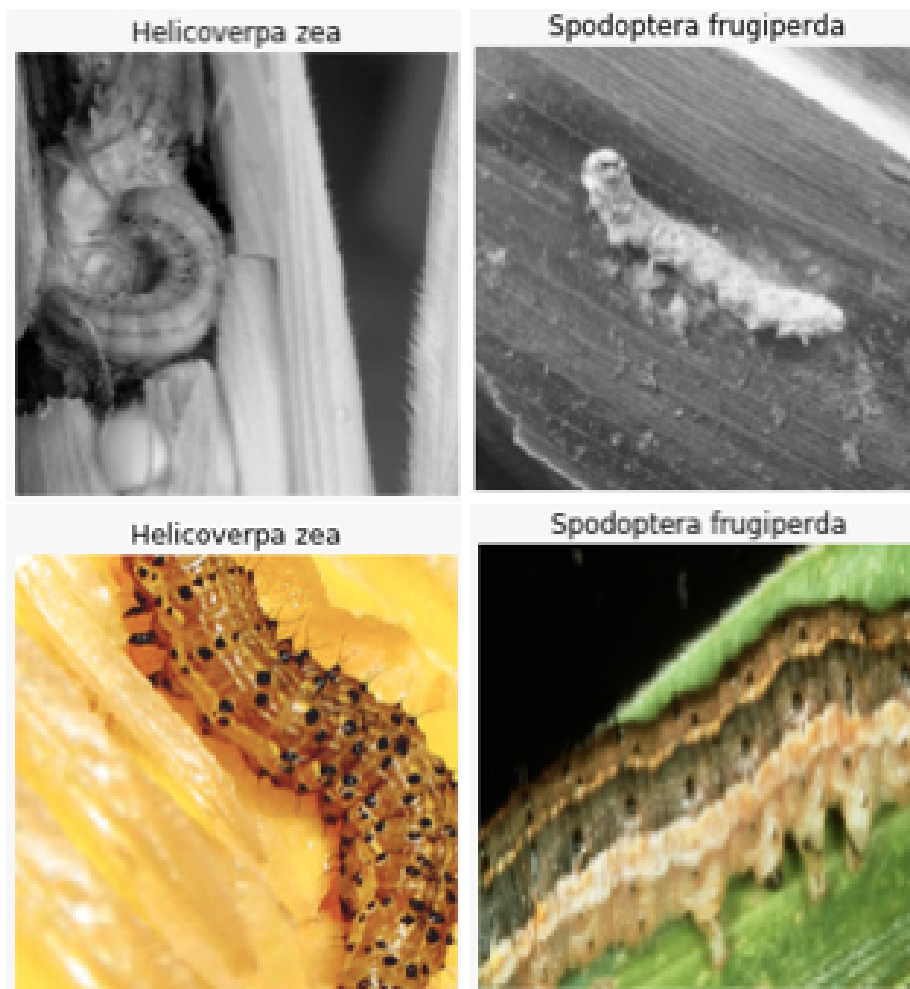
## 4.2 Configurações e Recursos dos Modelos de Aprendizagem

Os modelos de redes neurais convolucionais (RNCs) escolhidos para o experimento possuem parâmetros pré-treinados do conjunto ImageNet, no qual possui cerca de 1,5 milhão de imagens (DENG et al., 2009). Ambas as fases principais do experimento (primárias e todas) contaram com 720 imagens ao todo para cada categoria, subdivididos em aproximadamente  $\frac{560}{720} \times 100\% = 77,78\%$  para treinamento e  $\frac{160}{720} \times 100\% = 22,23\%$  para validação. Além disso, compuseram os modelos base para testes: um modelo SVM com extração de características binárias (*Local Binary Pattern - LBP*), um modelo de RNC direta (ALexNet) e um modelo de rede RNC residual (Inception-V3), os quais serviram de comparativo para o modelo de RNC modificado (Inception-V3\*).

Todas as redes RNCs treinadas foram configuradas com o otimizador de momento adaptativo (Adam) (KINGMA; BA, 2014) o qual é uma variação de descida estatística de gradiente de fator igual à taxa de aprendizagem. A função Adam é baseada em momento de descida de gradiente e tem duas variáveis para tal ( $\alpha$  e  $\beta$ ) as quais são obtidas por meio das médias móveis da descida de gradiente. Segundo a sugestão de Kingma, os valores para  $\alpha$  e  $\beta$  se encontram no intervalo entre 0,9 e 0,999 na maioria dos casos. O Algoritmo 1 contém todas os passos para realização do treinamento utilizados tanto para o experimento com as pragas primárias quanto para o treinamento com todas as pragas.



(a) Escala de cinza e sub-amostragem  $112 \times 112$ .



(b) Escala de cinza e tamanho normal  $224 \times 224$  colorida.

Figura 4.1 – Transformações alternativas de entrada: a) tamanho e cor b) cor.

É importante ressaltar que o modelo SVM com LBP utiliza configurações diferentes, devido às configurantes inerentes ao modelo como o fator de regularização ( $C$ ) descrito na Seção 2.3 igual a  $c = 100$ , raio de binarização do LBP ( $R$ )  $r = 8$  e número de vizinhos no raio de binarização ( $V$ ),  $v = 24$  com probabilidade de escolha ( $P$ ),  $p = 42\%$ .

Após a preparação dos passos com requisitos básicos como as dependências da biblioteca Fastai (HOWARD, 2018) e as bibliotecas de geração de resultado em gráficos (Matplotlib), passos seguintes estabelecem o caminho para o conjunto de imagens na máquina local, tamanho de imagem utilizado  $t = 224$ , tamanho de lote (*batch*)  $a = 32$  e o número de épocas iniciais  $b = 2$ . Em seguida, define-se otimizador de descida de gradiente (neste caso, Adam), a arquitetura da rede a ser treinada de acordo com as opções disponíveis e, por fim, cria-se o objeto de treinamento configurado com as variáveis anteriormente definidas e com os pesos ou parâmetros do ImageNet (DENG et al., 2009). Este objeto é simplesmente uma função da arquitetura escolhida e configurada com todas as variáveis necessárias para treinamento. O próximo passo no Algoritmo 1 refere-se à função de *Varredura* a qual foi acionada para determinar a taxa de aprendizagem inicial para o treinamento de cada rede. Esta função testa valores dentro do intervalo de  $10^{-22}$  a  $10^1$  os quais foram escolhidos empiricamente dentro um intervalo extrapolado de possíveis valores para taxas de aprendizagem. O objeto da função de *Varredura* é verificar se há um intervalo de descida de gradiente que termine no mínimo da função como visto na Figura 4.2, dentro do intervalo testado para a taxa de aprendizagem (HOWARD, 2018). Caso seja identificado o mínimo da função, escolhe-se o valor próximo a uma magnitude menor que aquele, configurando-se assim, a curva de descida de gradiente (HOWARD, 2018).

Uma vez identificado intervalo de descida de gradiente que leve ao mínimo da função de erro durante a varredura, inicia-se o treinamento o número inicial de épocas. Com o uso da biblioteca Fastai é possível guardar o estado atual da rede ao longo das interrupções entre épocas para que sejam efetuadas mais varreduras de taxa de aprendizagem. Com isso, é possível observar se a taxa de aprendizagem utilizada no intervalo de duas a quatro épocas anteriores ainda à curva de descida de gradiente que leve ao mínimo da função. Caso não haja mais uma curva de descida de gradiente aplica-se, opcionalmente, uma ampliação virtual no conjunto de validação, com intuito de verificar se tal curva reaparecerá, caso contrário encerra-se o treinamento e aplicam-se as avaliações finais ao modelo. As avaliações finais do modelo são uma validação cruzada com imagens virtuais aleatórias geradas por meio da ampliação do conjunto de validação.

Este capítulo abordou as etapas de treinamento com os detalhes de funcionamento de cada função, bem como o critério de definição dos valores iniciais de cada função. O Capítulo 5 apresenta os resultados obtidos para o treinamento de cada rede, por meio de tabelas comparativas, curvas de treinamento e validação e as respectivas matrizes de confusão para cada rede nas duas fases principais: primárias e todas.

---

**Algoritmo 1** Etapas Principais de Treinamento

---

**Data:**  $img_i$  tal que  $i = 0, 1, 2, \dots, N$  com  $N=560$

**begin**

*CarregarFastai, Matplotlib, seaborn, numpy, pandas;*

*bancoImagens*  $\leftarrow$  *CaminhoDiretorio;*

*tamanhoImagem*  $\leftarrow$  *t, batch*  $\leftarrow$  *a, epoca*  $\leftarrow$  *b;*

*Otimizador*  $\leftarrow$  *Adam( $\alpha, \beta$ );*

*Arquitetura*  $\leftarrow$  *Escolher(SVM, AlexNet, Inception - v3, Inception - V3\*)*

**if** *Arquitetura*  $==$  *SVM* **then**

*C*  $\leftarrow$  *c, R*  $\leftarrow$  *r;*

*V*  $\leftarrow$  *v, P*  $\leftarrow$  *p;*

*Rede*  $\leftarrow$

*parametros(bancoImagens, Arquitetura, tamanhoImagem, C, R, P);*

**end**

**else**

*Rede*  $\leftarrow$  *parametros(bancoImagens, Arquitetura, tamanhoImagem,*  
        *pesosImageNet, Otimizador);*

**end**

*EstatusTaxa, taxaInicial*  $\leftarrow$  *Algoritmo 2;*

**Function** *AmpliarValidacao(decisao)* *Validacao*  $\leftarrow$  *Validacao +*

*Rotacao(Validacao, 5%) + Zoom(Validacao, 10%) + Espelhar(10%)* **for**

*k*  $\in$  *Passos* **do**

**if** *EstatusTaxa*  $==$  *Verdadeiro* **then**

*epoca*  $\leftarrow$  *epoca - 1;*

*Parcial*  $\leftarrow$  *Treinar(Rede(taxaInicial, Epocas, batch, N, AmpVal));*

**end**

**else**

*taxa\_inicial*  $\leftarrow$  *Algoritmo 2;*

**if** *EstatusTaxa*  $\leftarrow$  *Falso* **then**

*AmpVal*  $\leftarrow$  *AmpliarValidacao(Verdadeiro);*

*break;*

**end**

**end**

*taxaInicial*  $\leftarrow$  *Algoritmo 2;*

*metricasValidacao*  $\leftarrow$  *validacao();*

*Erros*  $\leftarrow$  *Concatenar(Parcial, metricas\_validacao);*

**end**

*validacaoCruzada*  $\leftarrow$  *ValCruzada(Padrao = 8);*

*resultados*  $\leftarrow$  *Concatenar(validacaoCruzada, Fscore());*

**end**

---

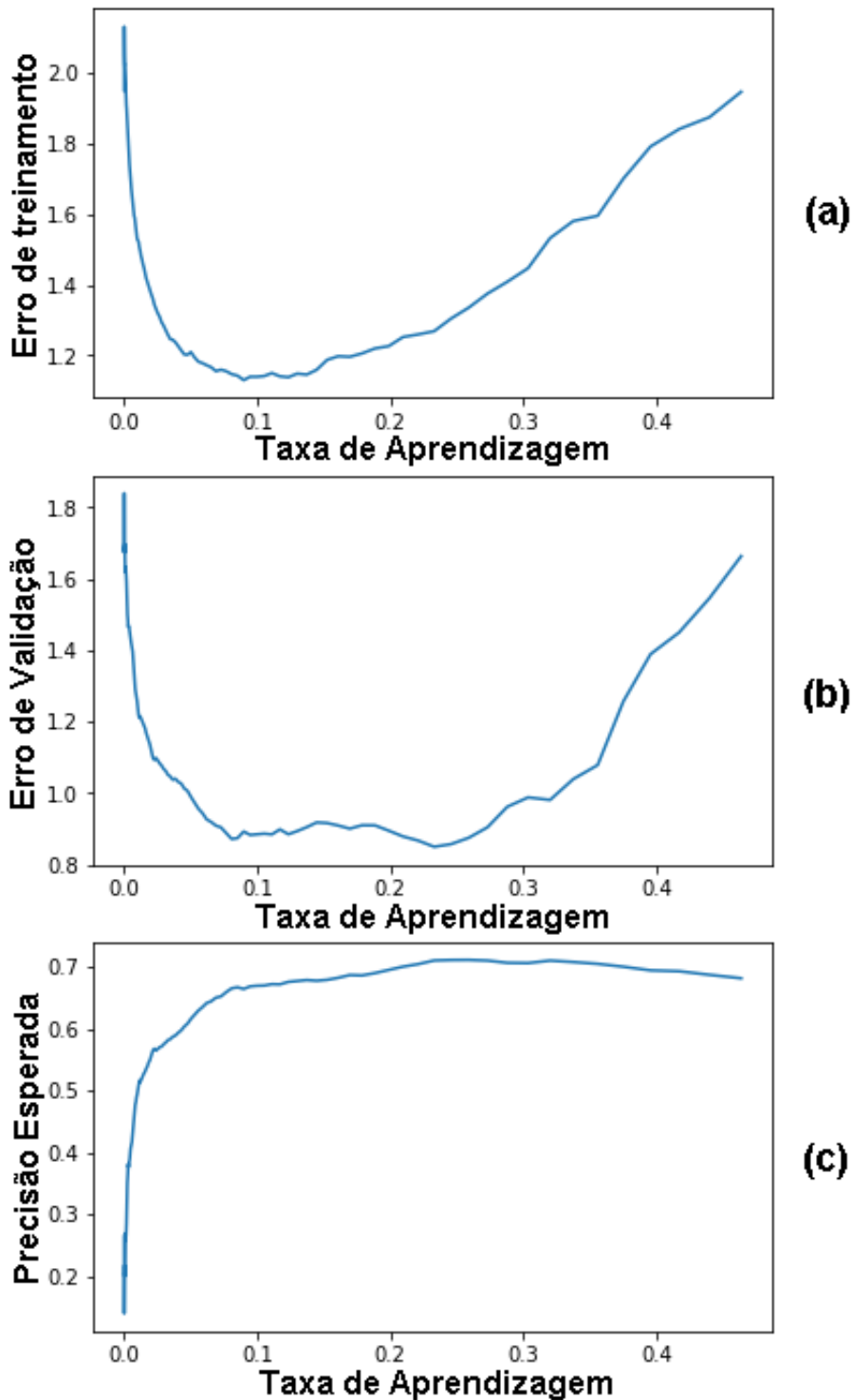


Figura 4.2 – Taxa inicial obtida através de varredura dentro do intervalo de  $10^{-22}$  a  $10^1$ , a) Erro de treinamento onde valor mínimo aponta o final do intervalo de descida de gradiente em ( $10^{-1}$ ) e o valor em uma magnitude abaixo de tal valor indica a taxa inicial, b) gráfico erro de validação para verificar se a taxa inicial escolhida atende também ao conjunto de validação, c) potencial de precisão atual para cada taxa no intervalo testado.

---

**Algoritmo 2** Algoritmo de Varredura de Taxa Inicial

---

```
begin
  Function Varredura(taxaInicial  $\leftarrow 10^{-22}$ , taxaFinal  $\leftarrow 10^1$ ,
    iteracoes  $\leftarrow 1000$ )
  for  $I \in$  iteracoes do
    | taxa  $\leftarrow$  Min(Treinar(taxaInicial, taxaFinal))/10;
  end
  if estatusTaxa = TreinaVel(taxaInicial) then
    | retorna(taxaInicial, Verdadeiro)
  end
  else
    | retorna(taxaInicial, Falso)
  end
  end;
end
```

---

## 5 Análise de Resultados

O Capítulo 5 apresenta os resultados obtidos pelo treinamento das redes neuronais clássica (SVM-LBP), RNCs direta (AlexNet) e residuais (Inception-V3 e Inception-V3\*). Destacou-se também o comportamento do modelo proposto (Inception-V3\*) frente às arquiteturas de base.

### 5.1 Desempenho dos modelos de Aprendizagem sobre o Conjunto de Imagens

Os resultados para o conjunto de pragas são avaliados a partir dos valores de precisão média da validação cruzada, curvas de erro e precisão, quantidade de épocas e desempenho individual por categoria testada.

#### 5.1.1 Resultados em Pragas Primárias

A Tabela 4 apresenta o desempenho de cada modelo para o conjunto de imagens das pragas primárias e imagens sem pragas. Os modelos RNC (AlexNet, Inception-V3 e o modelo proposto Inception-V3\*) foram substancialmente superiores ao SVM-LBP.

Tabela 4 – Desempenho médio (%) para cada modelo testado no conjunto de pragas primárias do milho e imagens sem pragas.

Categorias	SVM-LBP	AlexNet	Inception-V3	Inception-V3* Proposed
<i>Agrotis ipsilon</i>	28,1	98,1	97,5	<b>98,8</b>
<i>Diatraea saccharalis</i>	44,4	90,6	<b>92,5</b>	90,0
<i>Helicoverpa zea</i>	43,8	<b>98,8</b>	93,8	96,2
<i>Spodoptera frugiperda</i>	34,4	93,8	85,6	<b>96,9</b>
Sem Pragas	38,8	98,8	<b>100</b>	<b>100</b>
<b>Média</b>	37,9	96,0	93,8	<b>96,4</b>

O treinamento da redes RNCs variou entre 18 e 21 épocas, pois em torno da época 21 todas a RNCs alcançaram seu desempenho máximo, de forma após tal ponto, continuar o treinamento implica em degradação das redes e sobre-ajuste. Tais condições de



degradação se verificam através de um crescente erro de validação ao final de treinamento, o qual foi evitado nos testes. O valor de mini-lotes do conjunto de 32 foi aplicado às redes RNCs (AlexNet, Inception-V3 e Inception-V3\*). Ademais, os erros de treinamento e de validação foram monitorados juntamente com a precisão de ambos. A taxa inicial encontrada através da função *Varredura* descrita no Algoritmo 1, para todas as redes RNCs foi de  $10^{-2}$  com o mínimo inicial em  $10^{-1}$ .

A Tabela 4 mostra que o modelo Inception-V3\* obteve ligeira vantagem (96,4%) sobre a rede AlexNet (96,0%). Por outro lado, a rede Inception-V3\* mostrou-se 60% mais rápida que o modelo AlexNet para alcançar a sua precisão máxima possível para o conjunto testado. Enquanto, o modelo AlexNet demorou 25 épocas para atingir 96,0%, como visto na Figura 5.2, já a rede Inception-V3\* em apenas 10 épocas alcançou o seu melhor resultado como observado na Figura 5.1b. Quanto à comparação com o modelo Inception-V3, além de precisão 2.6% maior (96,4% para o modelo Inception-V3\* a 93,8% para a rede Inception-V3, visto na Figura 5.1c), o modelo Inception-v3\* foi 50% mais rápido.

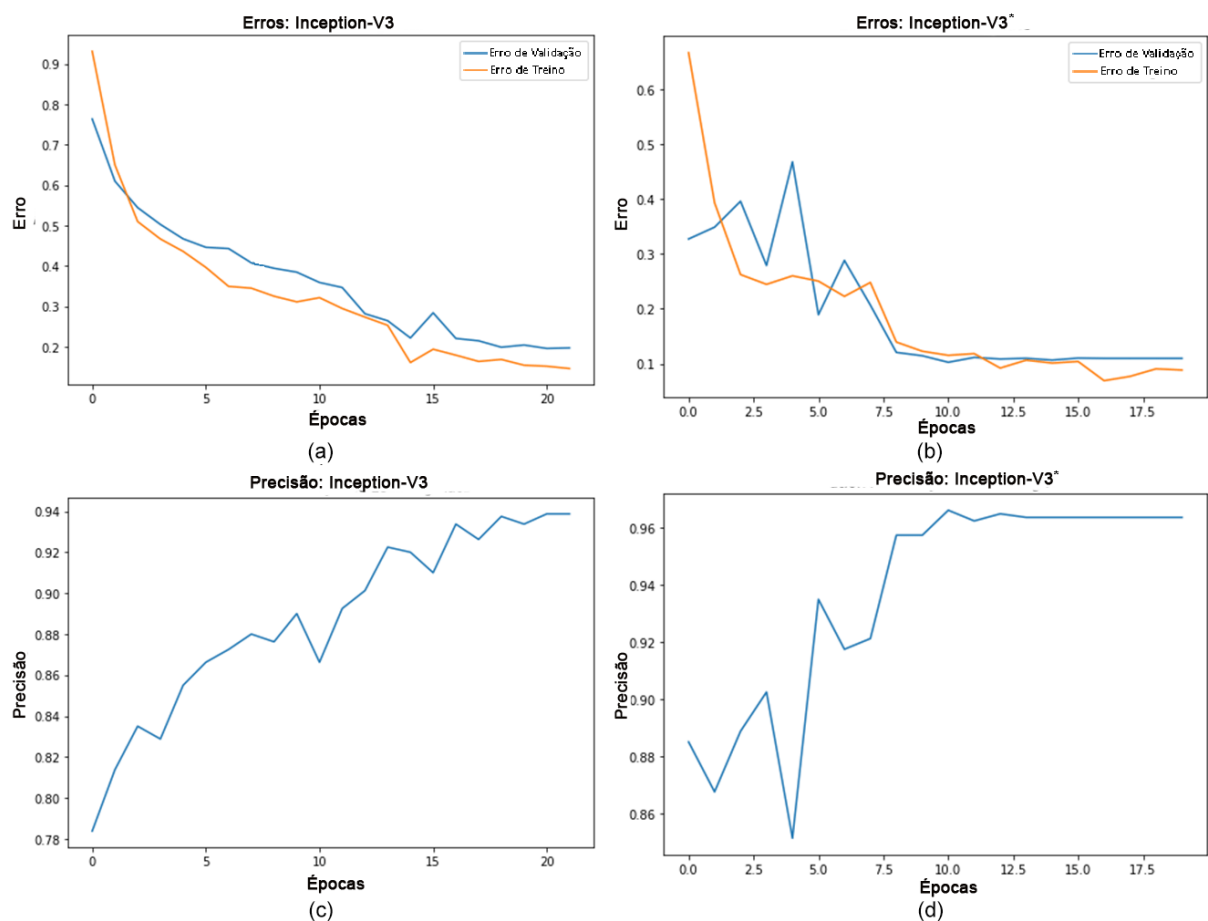


Figura 5.1 – Redução de erro e crescimento da precisão observados no primeiro experimento (Primárias). Para o modelo original Inception-V3, a) e c) respectivamente; e para o modelo proposto Inception-V3\* b) e d), respectivamente.

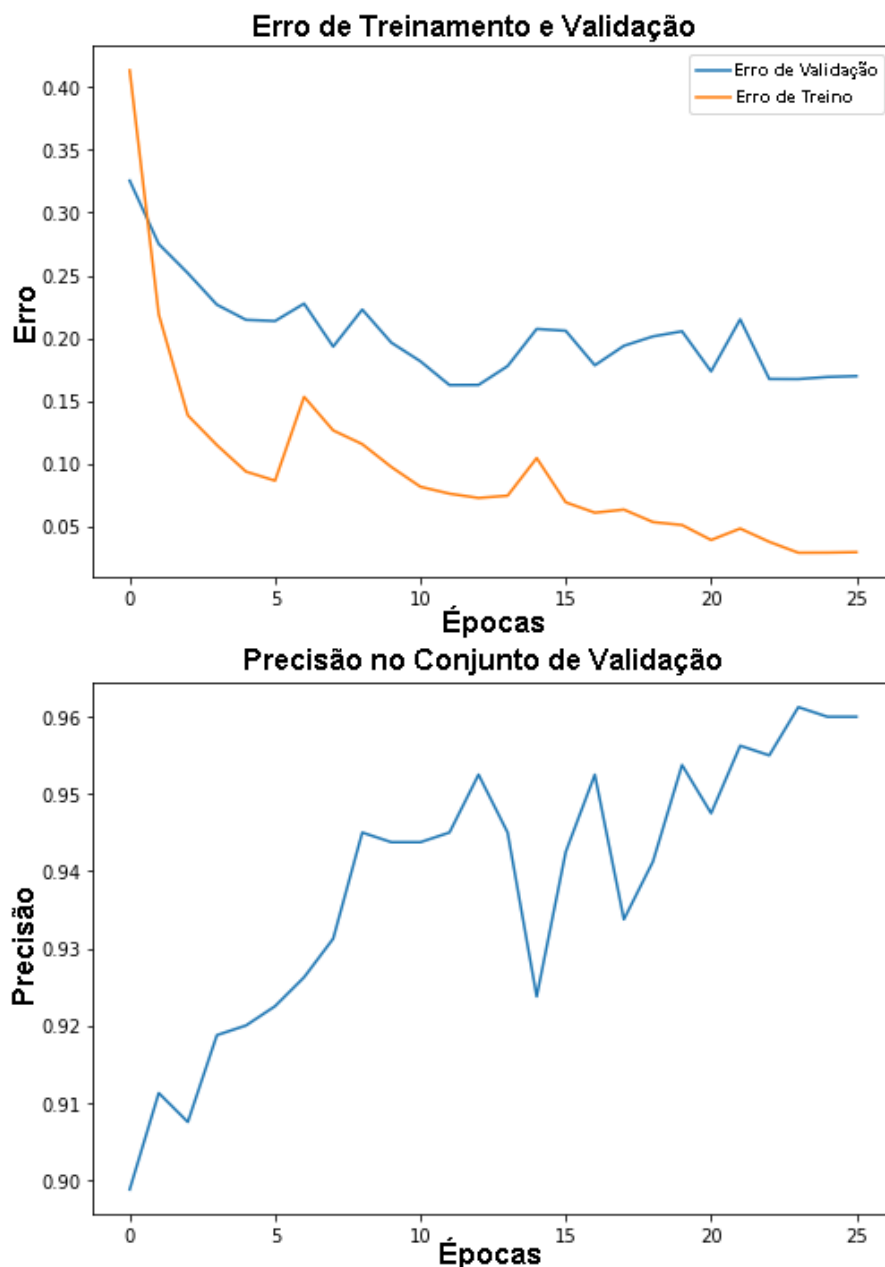


Figura 5.2 – Redução de erro e crescimento da precisão observados no primeiro experimento (Primárias) para o modelo AlexNet.

#### 5.1.1.1 Resultados individuais por categoria testada em Primárias- Matrizes de Confusão

O modelo SVM-LBP de precisão média de 37,9% já apresentada na Tabela 4, ficou abaixo de 45% em todas as cinco categorias do primeiro experimento. Entre as suas maiores taxas de acerto mostrados na Figura 5.3, destacam-se *Diatraea saccharalis* com 44,4%, majoritariamente confundido com *Helicoverpa zea* (21,2%) e *Spodoptera frugiperda* (21,9%) e, em seguida *Helicoverpa zea* com taxa de acerto de 43,8%, porém confundida de forma semelhante com as outras pragas e também com as imagens sem pragas.

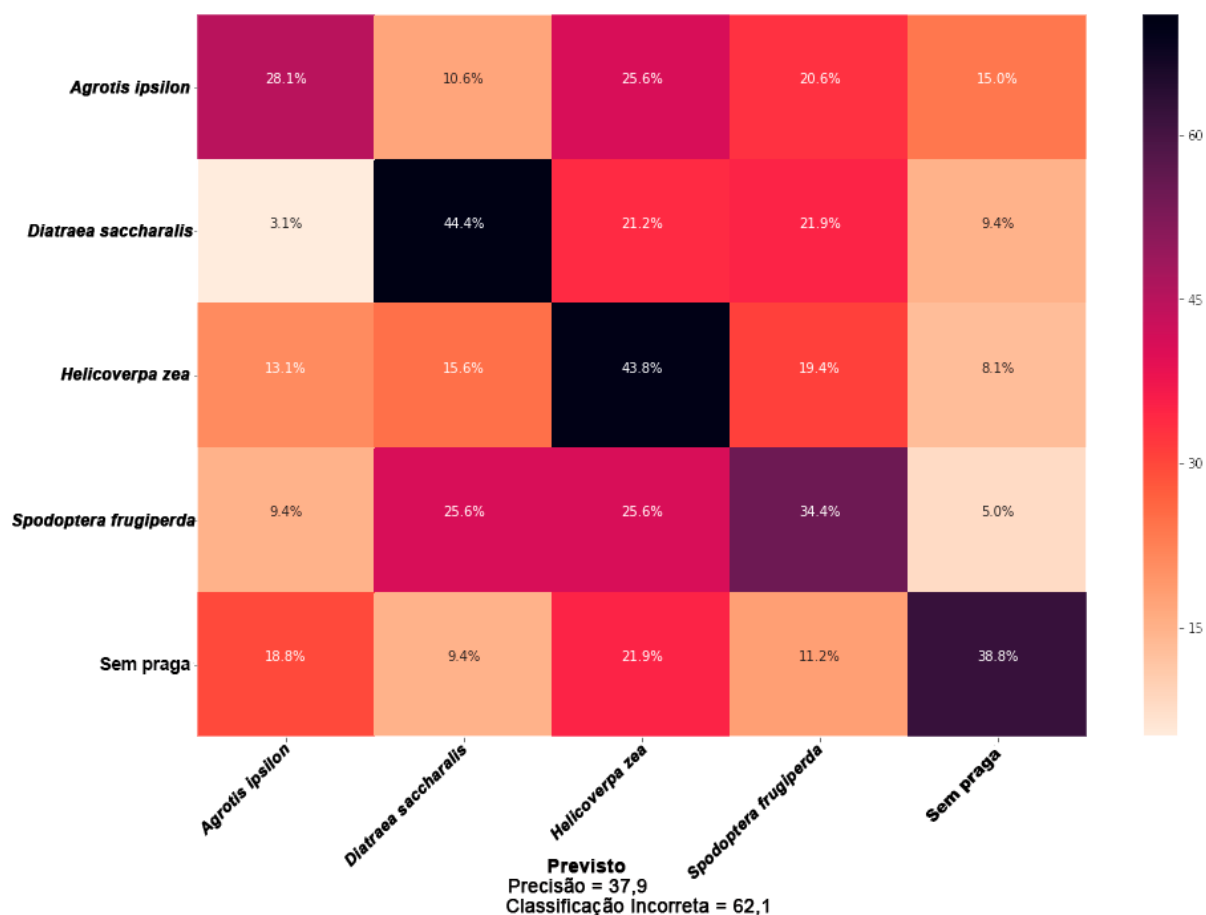


Figura 5.3 – Desempenho individual no conjunto de pragas primárias para o modelo SVM-LBP.

O modelo AlexNet apresentou excelente precisão média de 96,0% apresentada na Tabela 4 onde suas maiores dificuldades observadas na Figura 5.4 foram para o reconhecimento das pragas *Diatraea saccharalis* (90,6%) e *Spodoptera frugiperda* (93,8%) as quais apresentaram confusão mútua de 3,1%.

Em seguida, o modelo Inception-V3 alcançou bons resultados com precisão média de 93,8% conforme a Tabela 4. O modelo obteve 100% de precisão em imagens sem pragas, o que é relevante, pois demonstra a capacidade do modelo em distinguir o plano de fundo das lavouras das pragas estudadas. Por outro lado, a rede mostrou deficiências para distinguir a praga larval *Spodoptera frugiperda* (85,6%) ante as pragas *Diatraea saccharalis* e *Helicoverpa zea* e vice-versa (92,5% e 93,8% respectivamente) conforme observado na matriz de confusão da Figura 5.5.

Finalmente, o modelo proposto Inception-V3\* alcançou a maior precisão média de 96,4% conforme a Tabela 4. A Figura 5.6 mostra que, semelhante ao original, o modelo proposto obteve 100% de precisão em imagens sem pragas, mantendo assim, sua capacidade de distinção entre a presença e a ausência de pragas. Sua maior dificuldade foi o reconhecimento da praga larval *Diatraea saccharalis* (90%) a qual foi confundida com *Helicoverpa zea* (3,1%) e com *Spodoptera frugiperda* (6,2%). O modelo também apresentou melhores resultados para *Agrotis ipsilon* (98,8%), *Helicoverpa zea* (96,2%) e *Spodoptera*

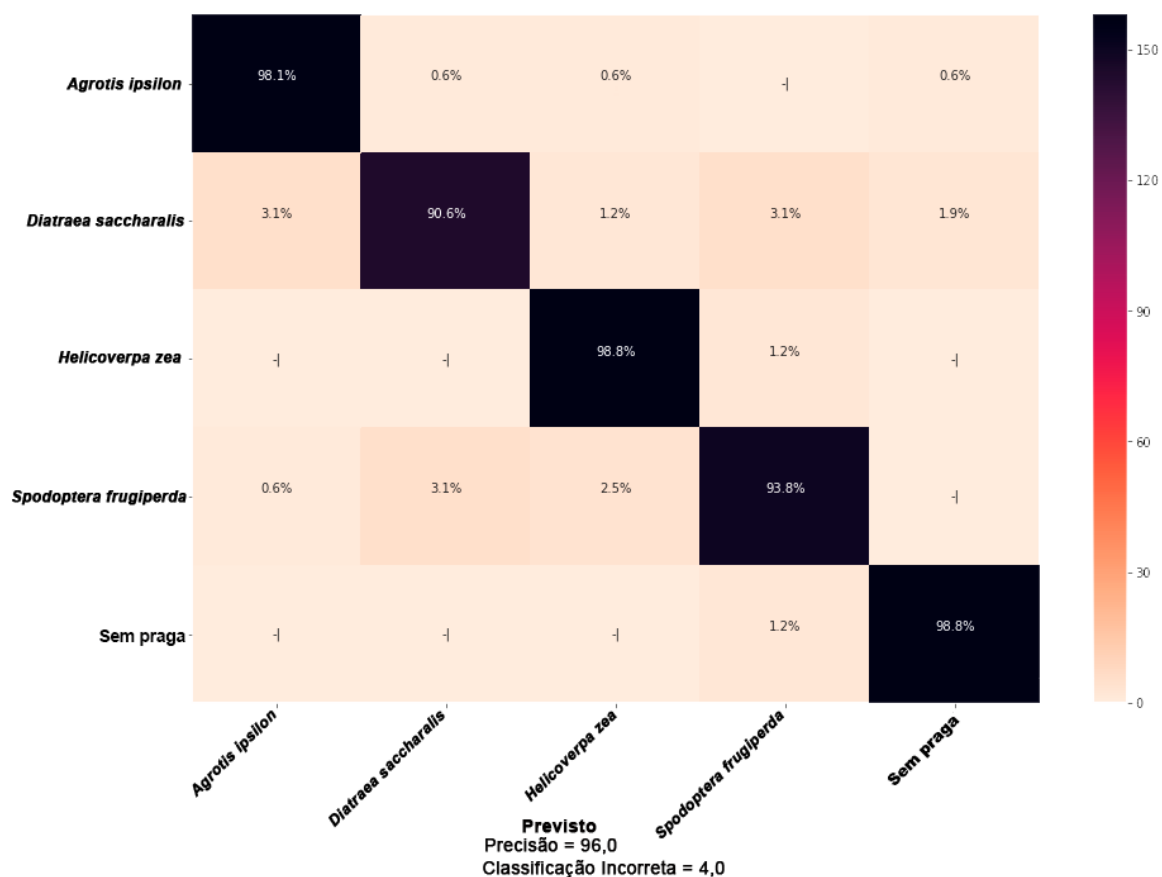


Figura 5.4 – Desempenho individual no conjunto de pragas primárias para o modelo AlexNet.

*frugiperda* (96,9%).

### 5.1.2 Resultados em todas as Pragas

No segundo experimento com também com tamanho fixo como no primeiro, pragas primárias, secundárias e imagens sem pragas foram testadas cujos resultados são expostos na tabela 5. Os resultados apontam ligeira vantagem para o modelo proposto Inception-V3\* o qual obteve precisão média de 97,0%, ALexNet em segundo com com 96,4% , seguido pela modelo de rede Inception-V3 com 95,0%. O modelo SVM-LBP apresentou resultados muito baixos com relação às redes RNCs alcançando apenas 42,0%. *Diatraea saccharalis* foi a espécie mais difícil para realização de reconhecimento pelos modelos convolucionais, ainda que o modelo regular Inception-V3 tenha alcançado 93,1% de precisão média.

Além disso, segundo as Tabelas 4 e 5 e Figuras 5.6 e 5.12, não somente no segundo, mas também no primeiro experimento, a espécie *Diatraea saccharalis* foi a mais difícil para o modelo proposto Inception-V3\*, o qual obteve um resultado abaixo de 90% (90% Primárias e 87.5% para todas). Por outro lado, o modelo proposto atingiu 100% de precisão nas pragas secundárias (*Dalbulus maidis* e *Diabrotica speciosa*). Quanto à praga *Spodoptera frugiperda*, o modelo proposto Inception-V3\* superou o modelo original em 11,3% no

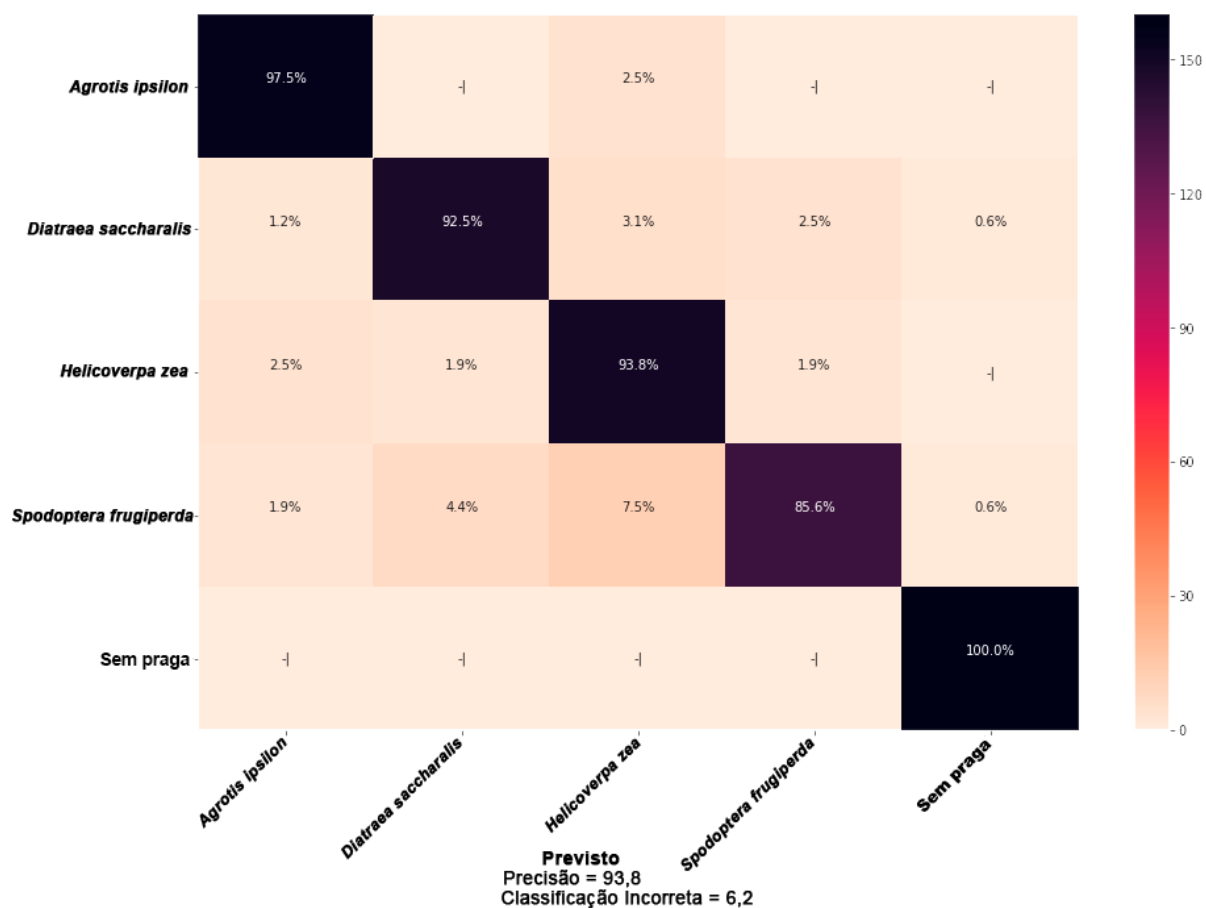


Figura 5.5 – Desempenho individual no conjunto de pragas primárias para o modelo Inception-V3.

primeiro experimento e, no segundo, quando todas as pragas foram testadas, em 12,5%.

Tabela 5 – Desempenho médio (%) para cada modelo testado com todas as pragas e imagens sem pragas.

Categorias Testadas	LBP-SVM	AlexNet	Inception V3	Proposed Inception-V3*
<i>Agrotis ipsilon</i>	32,5	97,6	97,5	<b>98,1</b>
<i>Diatraea saccharalis</i>	26,9	90,7	<b>93,1</b>	87,5
<i>Helicoverpa zea</i>	47,5	<b>97,0</b>	90,0	96,2
<i>Spodoptera frugiperda</i>	31,9	93,8	85,0	<b>97,5</b>
<i>Dalbulus maidis</i>	50,7	<b>100</b>	<b>100</b>	<b>100</b>
<i>Diabrotica speciosa</i>	55,0	96,3	99,4	<b>100</b>
Sem pragas	49,6	99,4	<b>100</b>	99,4
<b>Média</b>	42,0	96,4	95,0	<b>97,0</b>

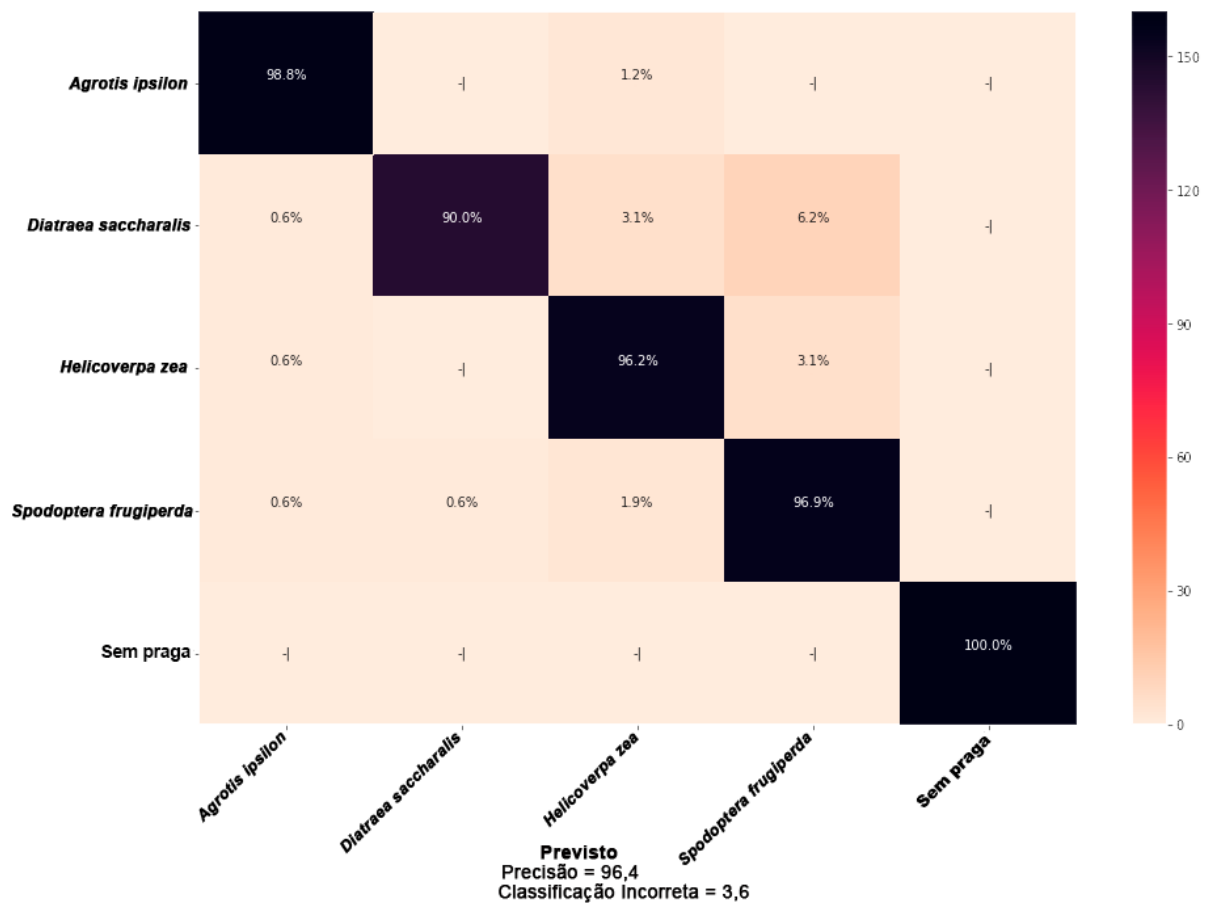


Figura 5.6 – Desempenho individual no conjunto de pragas primárias para o modelo Inception-V3\*.

Precisão e erros médios do modelo AlexNet obtidos no treinamento em todas as categorias (primárias, secundárias e sem pragas) são ilustrados nos gráficos da Figura 5.7. Este manteve sua quantidade de épocas necessárias para atingir seu melhor resultado, porém oscilou abaixo de 94% por pelo menos 14 épocas.

As Figuras 5.8, 5.11 e 5.12 ilustram uma comparação das curvas de erro e validação e, as respectivas matrizes de confusão dos modelos Inception-V3 e Inception-V3\* no experimento com todas as pragas. Assim, a maior precisão bem como aprendizagem mais rápida podem ser observadas em favor do modelo proposto Inception-V3\* com relação ao modelo original e também ao modelo AlexNet.

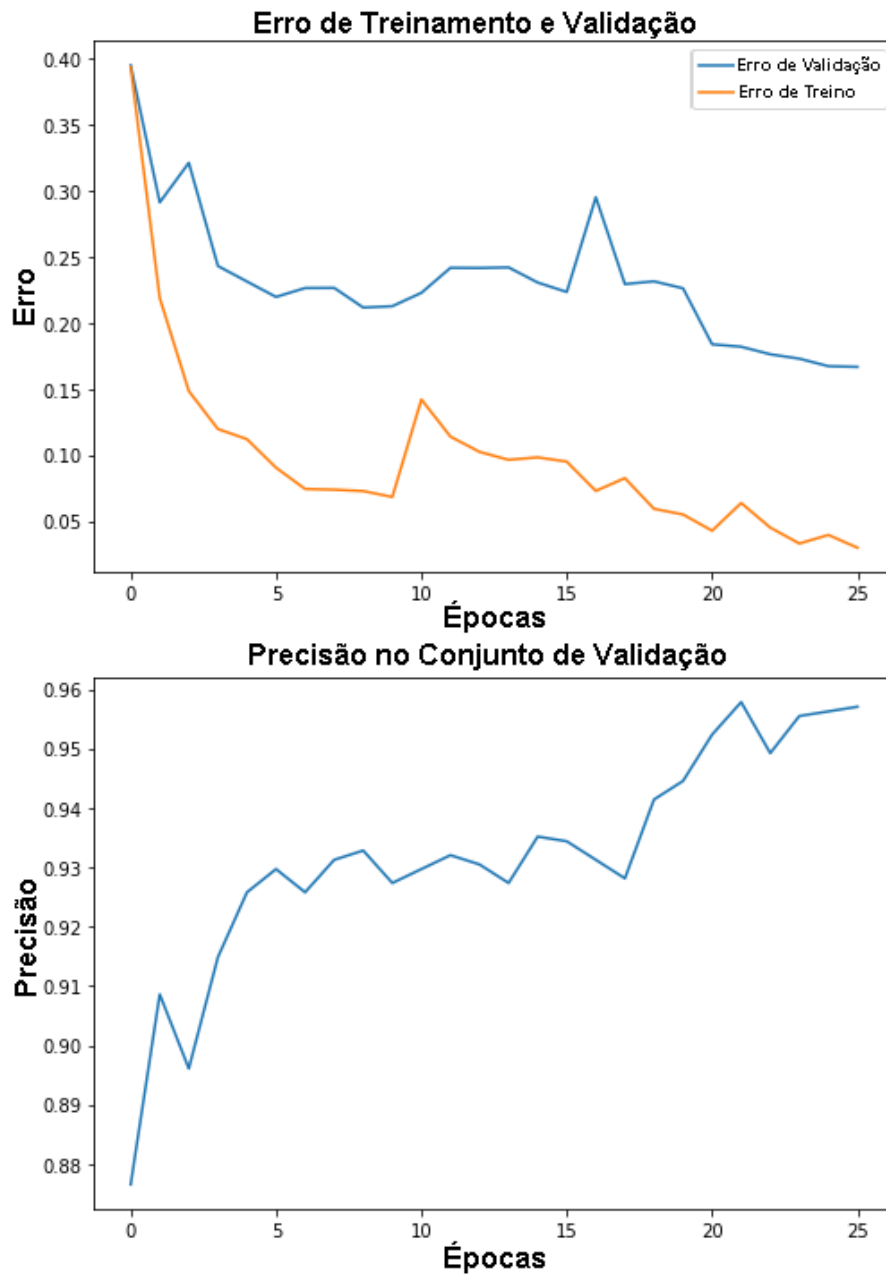


Figura 5.7 – Redução de erro e crescimento da precisão observados no primeiro experimento (Todas) para o modelo AlexNet.

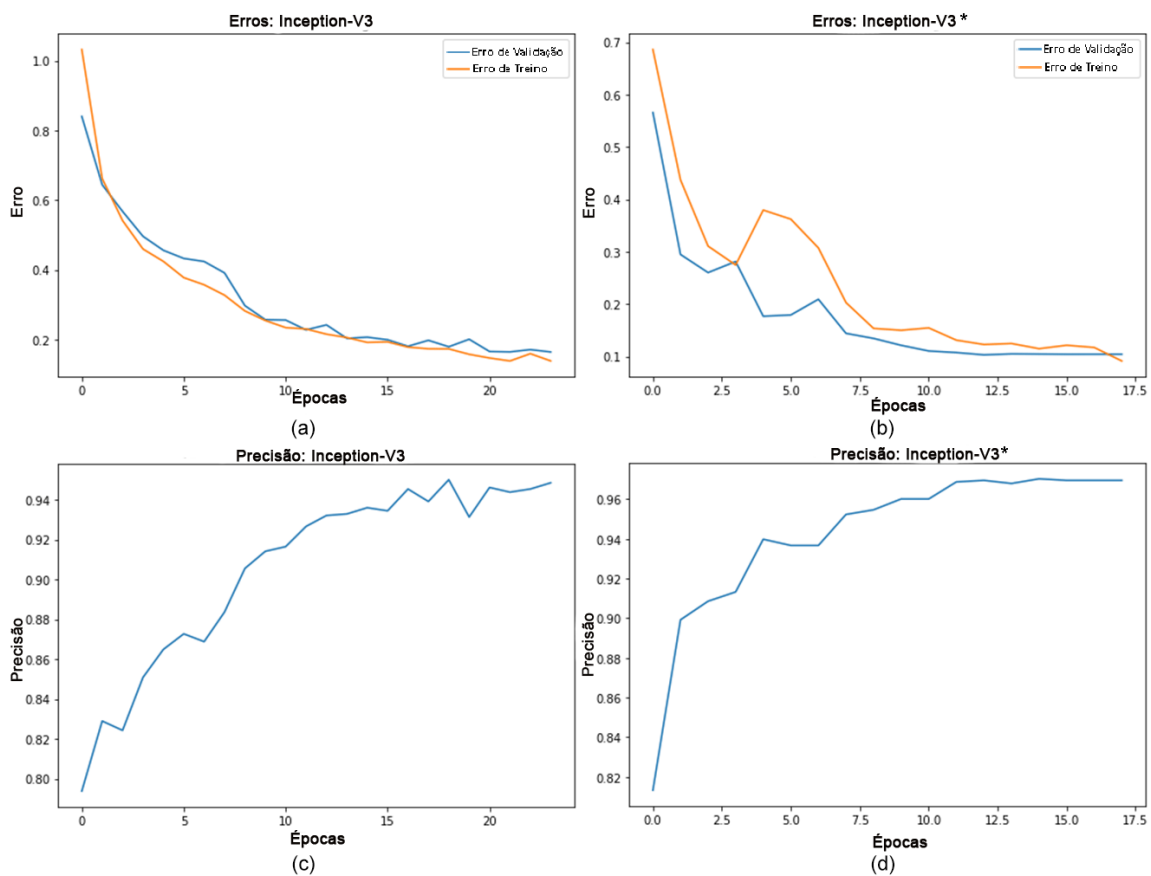


Figura 5.8 – Redução de erro e crescimento da precisão observados no primeiro experimento (Todas). Para o modelo original Inception-V3, a) e c) respectivamente; e para o modelo proposto Inception-V3\* b) e d), respectivamente.



### 5.1.2.1 Resultados individuais por categoria testada em Todas - Matrizes de Confusão

O modelo SVM-LBP cujos resultados são apresentados na Tabela 5 e Figura 5.9 obteve ligeira melhora para o teste com todas as pragas. Seus melhores resultados foram para as pragas secundárias, *Dalbulus maidis* e *Diabrotica speciosa*. Por outro lado, como o melhor resultado em 55%, ainda muito distante dos resultados para as redes RNCs.

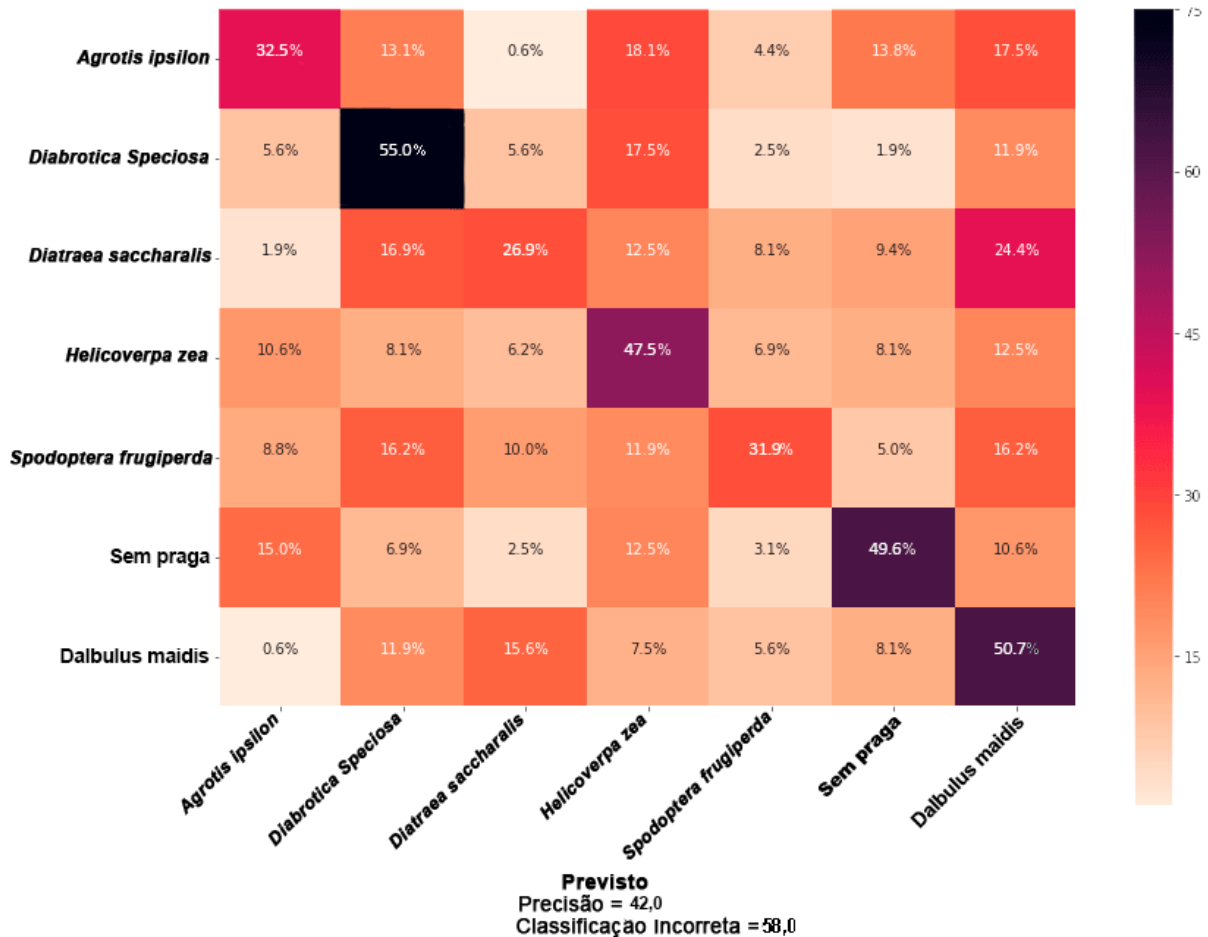


Figura 5.9 – Desempenho individual no conjunto de pragas (Todas) para o modelo SVM-LBP.

O modelo AlexNet também obteve desempenho ligeiramente superior ao obtido nas primárias, ou seja 0,4% a mais quando testado com todas as categorias, como se vê na Tabela 5 e Figura 5.10. A rede AlexNet obteve 100% em uma das pragas secundárias, *Dalbulus maidis*. A maior dificuldade do modelo foi em reconhecer a praga larval *Diatraea saccharalis* a qual foi confundida principalmente com outras pragas larvais e com imagens sem praga. Figura 5.10

Além de manter a plena capacidade de distinguir a presença ou não de pragas, como visto na Figura 5.11, o modelo Inception-V3 manteve-se também com dificuldades semelhantes para distinguir entre *Spodoptera frugiperda*, *Helicoverpa zea* e *Diatraea saccharalis*. Quanto às outras pragas, o desempenho da rede manteve-se próximo ao desempenho máximo.

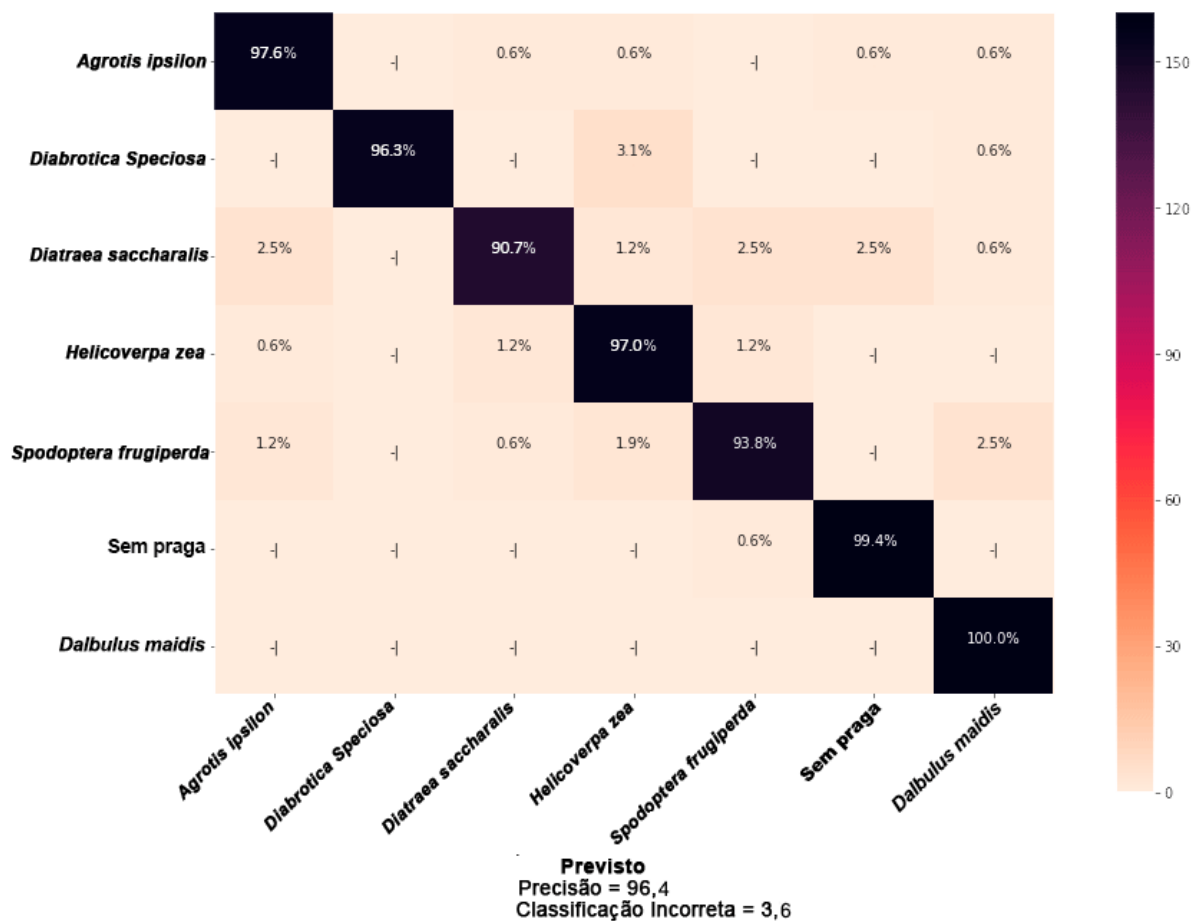


Figura 5.10 – Desempenho individual no conjunto de pragas (Todas) para o modelo AlexNet.

O modelo proposto alcançou novamente a maior média de precisão com 100% de precisão nas primárias como visto na Figura 5.12. Entretanto, houve maiores dificuldades para reconhecer a praga *Diatraea saccharalis*, principalmente com relação à *Spodoptera frugiperda*, apesar de ter sido um desafio difícil para todas as redes. Também, a rede alcançou 100% de precisão nas pragas secundárias e apenas uma das imagem sem pragas foi confundida com *Dalbulus maidis*.

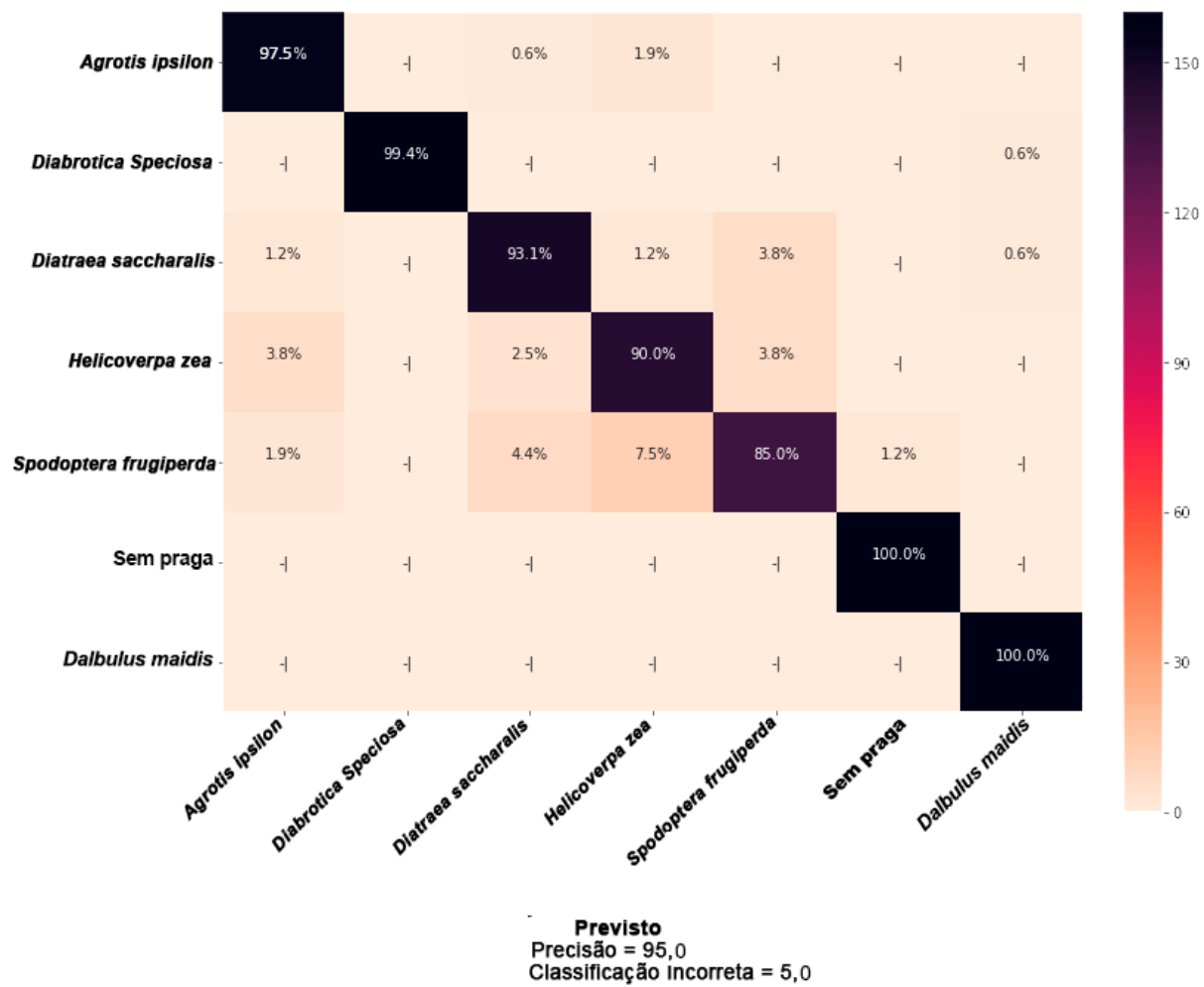


Figura 5.11 – Desempenho individual no conjunto de pragas (Todas) para o modelo Inception-V3.

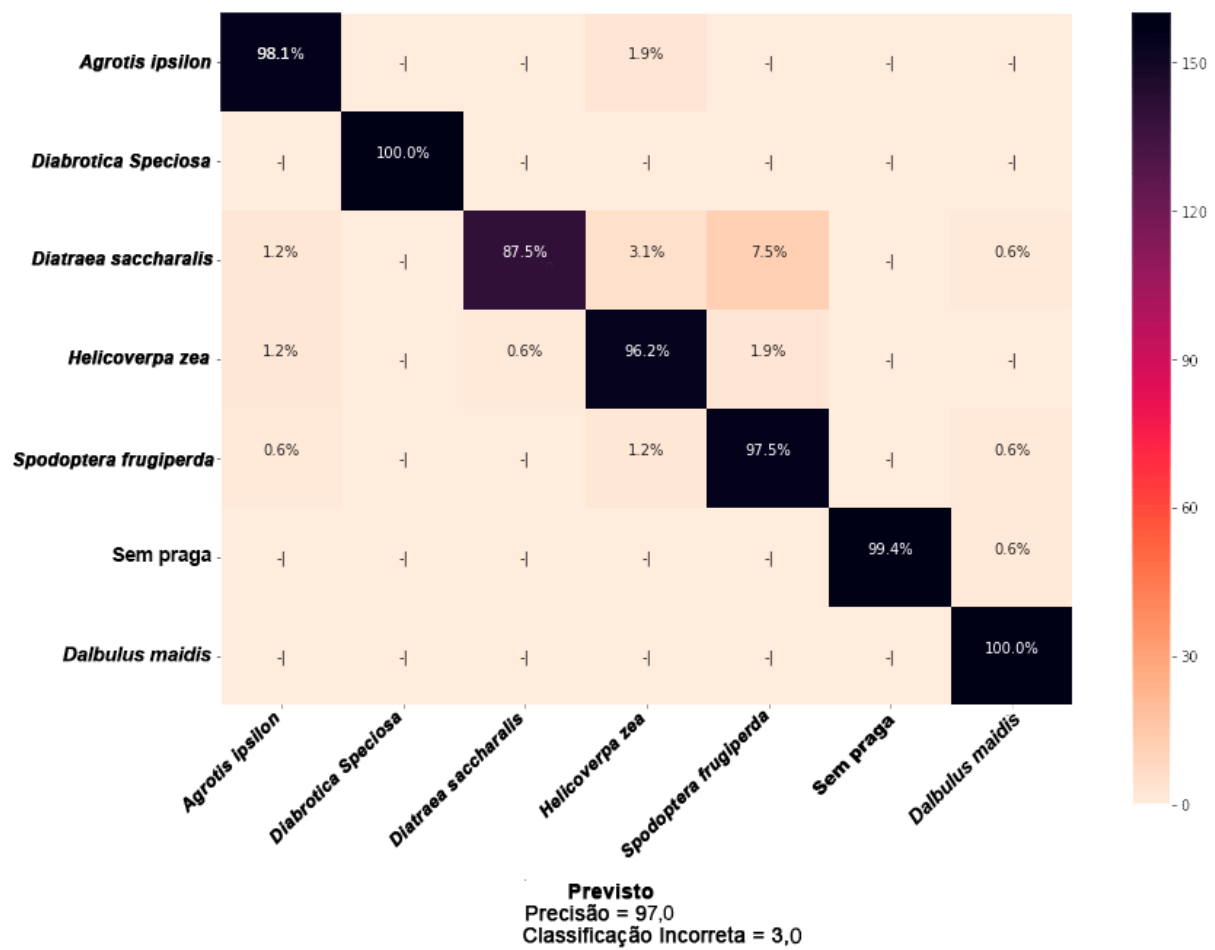


Figura 5.12 – Desempenho individual no conjunto de pragas todas para o modelo Inception-V3\*.

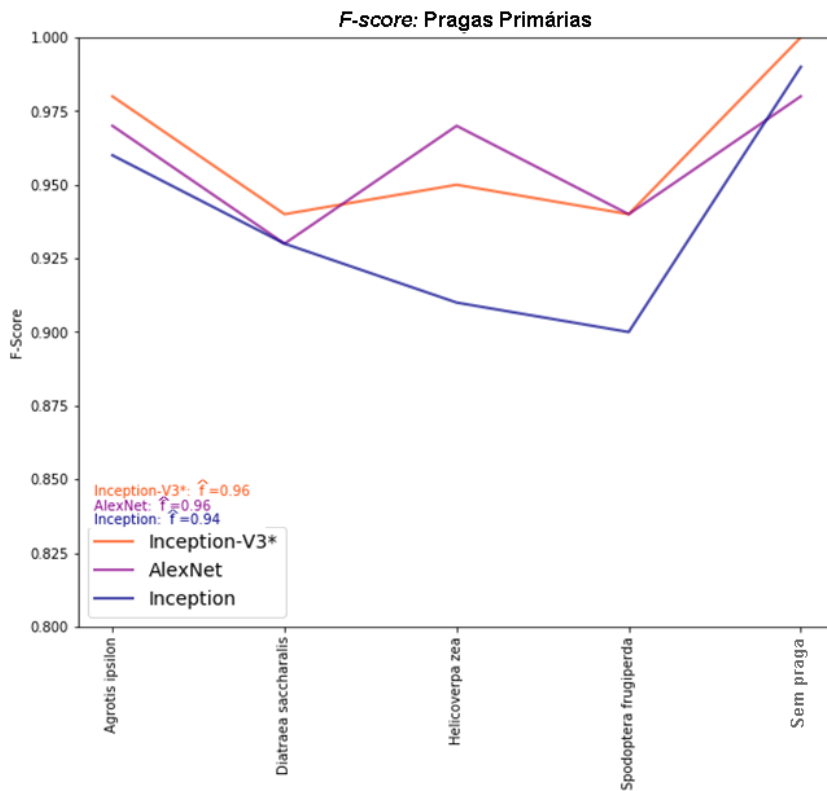
Por fim, as Figuras 5.13a e 5.13b apresentam todos os valores de  $F$ -score para ambas as fases do experimento contendo todas as redes RNCs, por obter os resultados mais expressivos. O modelo proposto Inception-V3\* manteve-se superior ou igualmente competitivo para 4 das 5 categorias no primeiro teste (primárias) e em 5 das 7 categorias no segundo teste (todas). Suas maiores dificuldades foram *Diatraea saccharalis* e *Helicoverpa zea* onde a Inception-V3 original e AlexNet alcançaram maiores precisões respectivamente, contudo, o modelo proposto manteve-se muito próximo e com  $F$ -score superior às duas para a *Diatraea saccharalis* no experimento com as primárias. Em média, o Inception-V3\* foi mais preciso, além de ser o mais rápido. Em conjuntos maiores e mais variados, tais diferenças podem ficar mais evidentes.

### 5.1.3 Comparativo de Desempenho da rede modificada no conjunto Original

Após os testes principais, foi realizado um teste para verificar se houve sobreajuste no conjunto ampliado por meio da comparação de seus resultados para uma rede com os resultados da mesma rede no conjunto original de imagens. No teste somente as imagens originais das 6 espécies de pragas e 1 categoria sem praga foram utilizadas, totalizando 700 imagens. Com esse isso, pode-se comparar seus resultados com aquele obtido pelo conjunto de dados de imagens ampliadas.

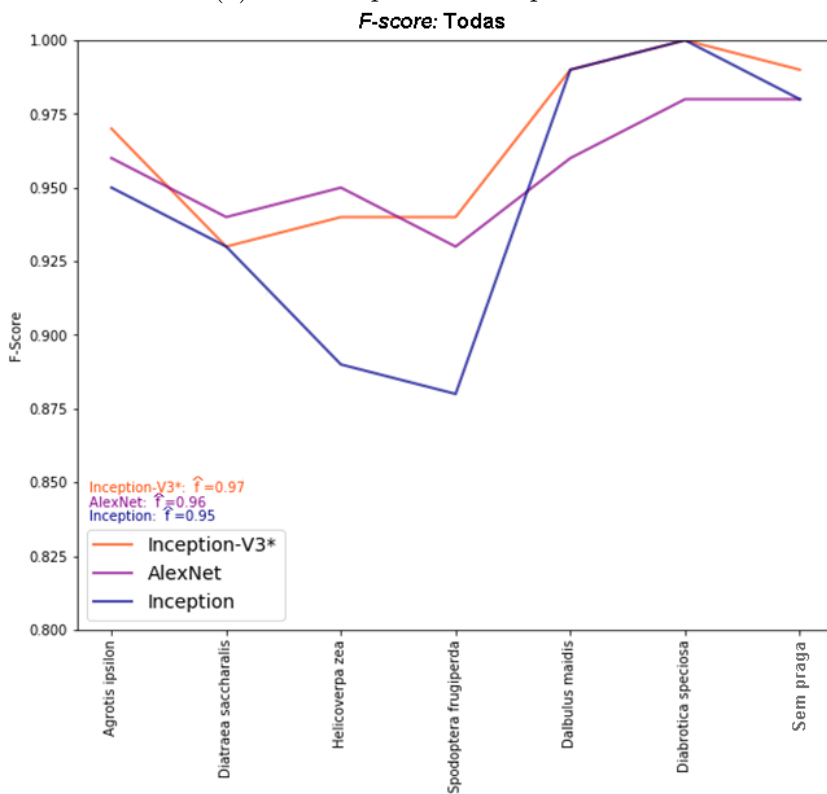
O comparativo foi realizado com a rede Inception-V3\* nos dois conjuntos. a Figura 5.14 sugere que o desempenho da rede Inception-V3\* para o conjunto original obteve comportamento próximo ao da rede testada sobre o conjunto ampliado, de forma que a diferença se dá pelo número de imagens o qual influencia as amplitudes dos erros de treinamento e de validação. Verifica-se também que suas posições em cada gráfico as quais estão invertidas, onde 5.14a onde o erro de treinamento se mantém superior ao de validação, devido às camadas de *dropout* e o menor número de imagens influenciar também tal comportamento.

Este Capítulo, portanto, abordou detalhes quanto aos resultados obtidos nas duas fases do experimento as quais mostraram que o modelo proposto foi mais substancialmente mais ágil e mais preciso em média. Individualmente, o modelo foi melhor em 80% da fase com as primárias e em 71% na fase com todas as pragas testadas. O Capítulo 6 expõe as considerações finais deste trabalho assim como possíveis contribuições para trabalhos futuros.



a)

(a) F-score para classes primárias.



b)

(b) F-score para todas as classes.

Figura 5.13 – Comportamento de cada modelo expresso pelos valores de *F-score* a) Pragas primárias b) Todas as pragas.

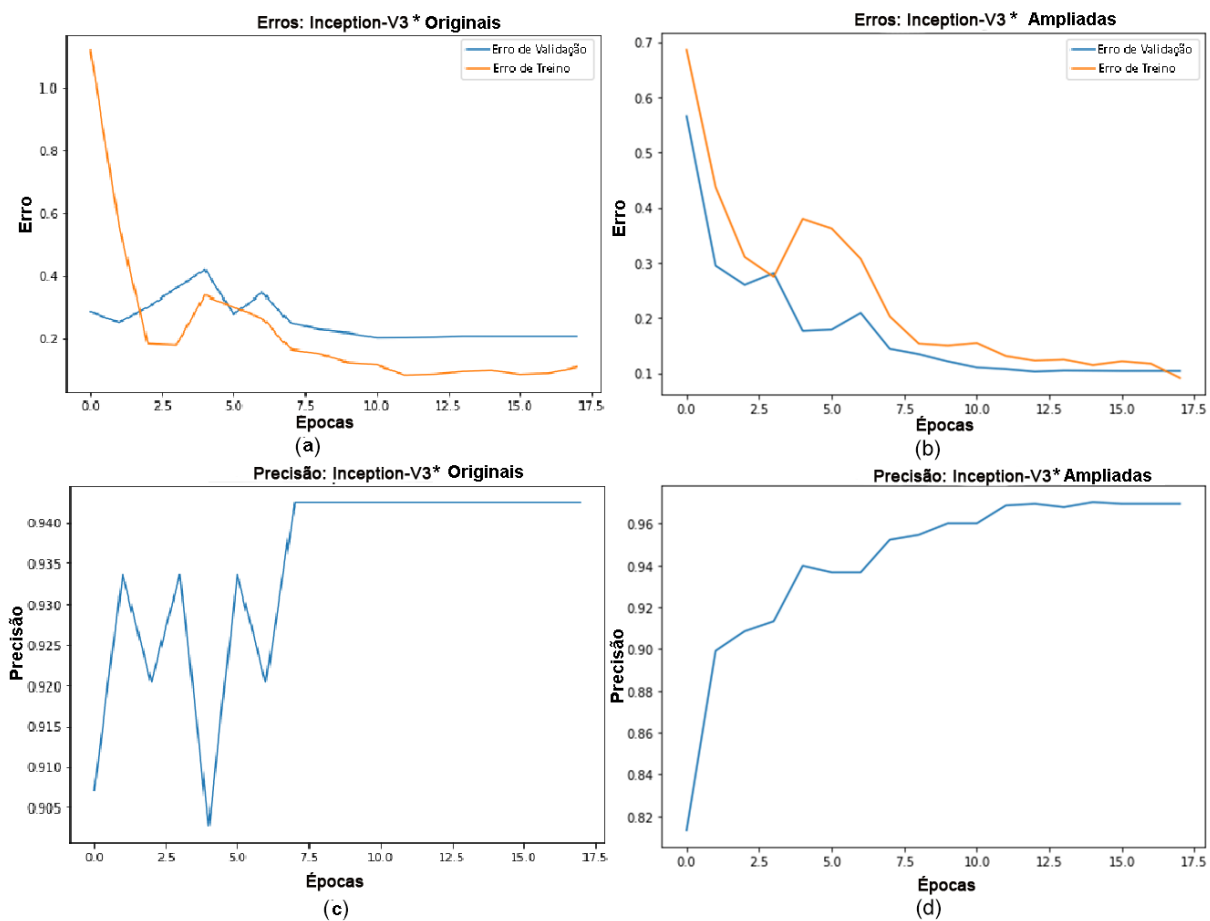


Figura 5.14 – Resultados Comparativos de erro e precisão da rede Inception-3\* nos conjuntos originais e ampliado: a) erros de treinamento e de validação (originais), b) precisão para o conjunto original de imagens, c) erros de treinamento e de validação (ampliadas), d) precisão no conjunto de imagens ampliadas.

## 6 Conclusões

Este estudo apresentou um conjunto inédito de imagens de campo para as pragas primárias e secundárias em milharais. Conjunto foi construído a partir de imagens tomadas em campo aberto e, ampliado em seguida por técnicas de processamento de imagem. O conjunto inicial contou com 700 imagens (600 de pragas e 100 de plantas sem a presença de pragas) que, após a ampliação, alcançou 5040 imagens, divididas proporcionalmente entre o conjunto de dados de treino e validação. As imagens de treino são expostas no Apêndice A para todas as categorias testadas.

Ademais, um teste comparativo de modelos de SVM, precedido por um extrator binário de características, modelos RNCs típicos (AlexNet), e residuais (Inception-V3 e a versão Proposta Inception-V3\*). A série de pré-processamento aplicado no banco de imagens, juntamente com os recursos de transferência de aprendizagem, otimizadores, e os recursos pré-processamento das redes, disponíveis biblioteca Fastai, os quais geram imagens extra durante a validação proporcionaram maior precisão às redes. A combinação dos recursos aplicados em tempo propício e de maneira correta, durante o treinamento, fez com que, por exemplo, as redes obtivessem maior desempenho, com destaque para os modelos Inception-V3, AlexNet e a proposição seguiram respectivamente 95,0% , 96,4% e 97,0%.

O reconhecimento de pragas de lavouras é um desafio complexo, porém pode contribuir muito para conduzir o manejo integrado de pragas a novos patamares de eficiência bem como na redução do uso de pesticidas, estabelecendo-se assim, um ecossistema simplificado com maior grau de sustentabilidade. Embora este seja um dos maiores conjuntos de imagens de pragas para *Zea mays ssp.* aplicado ao problema de reconhecimento de suas pragas primárias e secundárias, o conjunto pode ser estendido a fim de cobrir outras situações de reconhecimento mais complexas e desafiadoras dentro do problema. Para se alcançar as condições de teste para tais situações mais complexas, deve-se coletar mais imagens das pragas estudadas em poses e variações que o pré-processamento não pode construir, bem como a inclusão de outras pragas não abordadas devido a sua baixa disponibilidade.

Um terceiro experimento, além dos dois realizados neste trabalho (treinamento com pragas primárias e treinamento com todas as pragas), pode-se também aproveitar a aprendizagem obtida no treinamento com as primárias e realizar uma transferência



de aprendizagem de transição para o segundo experimento. Assim, há possibilidade de comparar os desempenhos das redes quanto aos dois treinamentos isolados a partir da transferência de aprendizagem do *ImageNet* com um treinamento contínuo, onde o que foi aprendido com as primárias seria transferido ao treinamento com todas as pragas.

Há também dentro da população de insetos e outras classes de animais de escalas próximas de tamanho (ácaros, moluscos, aracnídeos), os quais são recorrentes no ecossistema de milharais e exercem outras funções deferentemente das pragas. Por exemplo, predadores de outros insetos, e outros insetos inofensivos à plantação podem compor categorias adicionais do conjunto de imagens. Tal composição com outros insetos e pequenos animais residentes no ecossistema podem corroborar para aferição da capacidade das redes testadas neste trabalho. Por fim, não somente o reconhecimento pode ser obtido, mas também a localização dos alvos dentro das imagens seria outra contribuição adicional tanto para o propósito de localização em si, quanto para a ratificação da capacidade das redes quanto a distinção entre um alvo na cena e seu plano de fundo.

# REFERÊNCIAS BIBLIOGRÁFICAS

- BLAKE, M. *Maize for the Gods: Unearthing the 9,000-year History of Corn*. Oakland, California: Univ of California Press, 2015.
- CARVALHO, F. P. Pesticides, environment, and food safety. *Food and Energy Security*, Wiley Online Library, v. 6, n. 2, p. 48–60, 2017.
- CHENG, X. et al. Pest identification via deep residual learning in complex background. *Computers and Electronics in Agriculture*, Elsevier, v. 141, p. 351–356, 2017.
- CRUZ, I. et al. Cultivo do milho: pragas da fase vegetativa e reprodutiva. *Embrapa Milho e Sorgo-Comunicado Técnico (INFOTECA-E)*, Sete Lagoas: Embrapa Milho e Sorgo, 2002., 2002.
- DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. Miami Beach, Flórida, 2009. p. 248–255.
- EBRAHIMI, M. et al. Vision-based pest detection based on svm classification method. *Computers and Electronics in Agriculture*, Elsevier, v. 137, p. 52–58, 2017.
- ESPINOZA, K. et al. Combination of image processing and artificial neural networks as a novel approach for the identification of bemisia tabaci and frankliniella occidentalis on sticky traps in greenhouse agriculture. *Computers and Electronics in Agriculture*, Elsevier, v. 127, p. 495–505, 2016.
- FAITHPRAISE, F. et al. Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters. *International Journal of Advanced Biotechnology and Research*, BioIT InternationalsJournals, v. 4, n. 2, p. 189–199, 2013.
- FAOSTAT. *FAOSTAT data, 2017*. 2019. <<http://faostat.fao.org>>. Acesso em 15 de abril de 2019.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. São Paulo, SP: Edgard Blucher, 2000.
- GOODFELLOW, I. et al. *Deep learning*. Cambridge, UK: MIT press, 2016.
- GOODFELLOW, I. et al. Generative adversarial nets. In: NIPS. *Advances in neural information processing systems*. Montreal, Canada, 2014. p. 2672–2680.
- HE, K. et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 37, n. 9, p. 1904–1916, 2015.

- HE, K. et al. Deep residual learning for image recognition. In: IEEE. *Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas, Nevada, 2016. p. 770–778.
- HOWARD, e. a. J. *The fast.ai deep learning library, lessons, and tutorials*. 2018. <<https://github.com/fastai/fastai,2018>>.
- KALCHBRENNER, N. et al. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.
- KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, Elsevier, v. 147, p. 70–90, 2018.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- KRIZHEVSKY, A. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.
- KRIZHEVSKY, A. et al. Imagenet classification with deep convolutional neural networks. In: NIPS. *Advances in neural information processing systems*. Lake Tahoe, Nevada, 2012. p. 1097–1105.
- LECUN, Y. et al. Deep learning. nature 521. *Nature*, v. 521, p. 436–444, 2015.
- LIM, S. et al. Performance effect analysis for insect classification using convolutional neural network. In: IEEE. *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSC)*. Penang, Malaysia, 2017. p. 210–215.
- LIU, Z. et al. Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. *Scientific reports*, Nature Publishing Group, v. 6, p. 20410, 2016.
- MAHARLOOEI, M. et al. Detection of soybean aphids in a greenhouse using an image processing technique. *Computers and electronics in agriculture*, Elsevier, v. 132, p. 63–70, 2017.
- MARTINEAU, M. et al. A survey on image-based insect classification. *Pattern Recognition*, Elsevier, v. 65, p. 273–284, 2017.
- MAYO, M.; WATSON, A. T. Automatic species identification of live moths. In: SPRINGER. *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Cambridge, UK, 2006. p. 58–71.
- MEDEIROS, M. A. d. *rincípios e práticas ecológicas para o manejo de insetos-praga na agricultura / Maria Alice de Medeiros... [et al.]*. Brasília, 2010.
- MOREIRA, H. J. da C.; ARAGÃO, F. D. *Manual de Pragas do Milho*. Campinas-SP: Agrolink, 2009.
- MUNDADA, R. G.; GOHOKAR, V. Detection and classification of pests in greenhouse using image processing. *IOSR Journal of Electronics and Communication Engineering*, v. 5, n. 6, p. 57–63, 2013.
- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: ACM. *Proceedings of the 27th international conference on machine learning (ICML-10)*. Haifa, Israel, 2010. p. 807–814.

- OECD. *Crop production*. 2018. <<https://doi.org/10.1787/49a4e677-en>>. Acesso em 12 de abril de 2019.
- SANTURKAR, S. et al. How does batch normalization help optimization? In: NIPS. *Advances in Neural Information Processing Systems*. Montreal, Canada, 2018. p. 2483–2493.
- SCHNEIDER, A. et al. *Manual de pragas 2015 Monsanto Brasil*. São Paulo, SP, 2015.
- SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.
- SUGAYAMA, R. et al. Defesa vegetal: fundamentos, ferramentas, política e perspectivas. belo horizonte: Sociedade brasileira de defesa agropecuária, 2015. 544p. sutherst, rw pest species distribution modelling: origins and lessons from history. *Biological Invasions*, v. 16, p. 239–256, 2014.
- SZEGEDY, C. et al. Going deeper with convolutions. In: IEEE. *Proceedings of the IEEE conference on computer vision and pattern recognition*. Boston, Massachusetts, 2015. p. 1–9.
- SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: IEEE. *Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas, Nevada, 2016. p. 2818–2826.
- USDA. *Grain: World Markets and Trade 2019*. 2019. United States Department of Agriculture Foreign Agricultural Service. Acesso em 2 de maio de 2019.
- USDA. *World Agricultural Production*. 2019. United States Department of Agriculture Foreign Agricultural Service. Acesso em 7 de maio de 2019.
- VALICENTE, F. H. Manejo integrado de pragas na cultura do milho. *Circular Técnica*, v. 208, p. 1–13, 2015.
- VASA, H. *Google Images Download*. 2018. <<https://github.com/hardikvasa/google-images-download,2018>>.
- XIE, C. et al. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Computers and Electronics in Agriculture*, Elsevier, v. 119, p. 123–132, 2015.

# Apêndices

# A Amostras de imagens da base de classes de insetos

O Apêndice A Contém exemplos das imagens de treino utilizadas nas fases do experimento com pragas primárias e secundárias, além das imagens sem pragas. Na primeira fase 4 pragas: *Agrotis ipsilon*, *Diatraea saccharalis*, *Helicoverpa zea*, *Spodoptera frugiperda*, além de imagens sem pragas. Na segunda fase foram utilizadas todas as pragas (6 no total), onde duas são secundárias: *Diabrotica speciosa* e *Dalbulus maidis*, além de imagens sem pragas (Figura A.1g).

## A.1 Imagens de Treinamento: Pragas Primárias

O conjunto de pragas primárias é composto 4 espécies de pragas testadas com 560 imagens cada uma, em um total de 2240 imagens de pragas, além de 560 imagens pra a categoria sem pragas. Compõem as pragas primárias: *Spodoptera frugiperda* (Figura A.1a), *Helicoverpa zea* (Figura A.1b), *Diatraea saccharalis* (Figura A.1c), *Agrotis ipsilon* (Figura A.1d).

## A.2 Imagens de Treinamento: Secundárias

O conjunto de pragas secundárias é composto de 2 espécies de pragas testadas com 560 imagens cada uma, em um total de 1120 imagens de pragas. As pragas secundárias são: *Diabrotica speciosa* (Figura A.1e) e *Dalbulus maidis* (Figura A.1f).



Figura A.1 – Exemplo de imagens de treinamento:(a) *Agrotis ipsilon*, (b) *Diatraea saccharalis*, (c) *Helicoverpa zea*, (d) *Spodoptera frugiperda*, (e) *Diabrotica speciosa*, (f) *Dalbulus maidis*, (f) imagens sem pragas.