

UNIVERSIDADE ESTADUAL DE PONTA GROSSA
SETOR DE CIÊNCIAS AGRÁRIAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

RENAN DE PAULA ROSA

**MÉTODO DE CLASSIFICAÇÃO DE PRAGAS POR MEIO DE REDE NEURAL
CONVOLUCIONAL PROFUNDA**

**PONTA GROSSA
2018**

RENAN DE PAULA ROSA

**MÉTODO DE CLASSIFICAÇÃO DE PRAGAS POR MEIO DE REDE NEURAL
CONVOLUCIONAL PROFUNDA**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada, curso de Mestrado em Computação Aplicada da Universidade Estadual de Ponta Grossa, como requisito parcial para obtenção do título de Mestre.

Orientadora: Prof^a. Dr^a. Maria Salete Marcon
Gomes Vaz

**PONTA GROSSA
2018**

R788 Rosa, Renan de Paula
Método de classificação de pragas por meio de rede neural convolucional profunda/ Renan de Paula Rosa. Ponta Grossa, 2018. 99 f.; il.

Dissertação (Mestrado Computação Aplicada) - Universidade Estadual de Ponta Grossa.

Orientadora: Profa. Dra. Maria Salete Marcon Gomes Vaz

1. Machine Learning. 2. Deep Learning. 3. Rede neural convolucional. 4. Classificação de imagens. I. Vaz, Maria Salete Marcon Gomes. II. Universidade Estadual de Ponta Grossa. Mestrado em Computação Aplicada. III. T.

CDD : 004.3


TERMO DE APROVAÇÃO


Renan de Paula Rosa

**“MÉTODO DE CLASSIFICAÇÃO DE PRAGAS POR MEIO DE REDE NEURAL
CONVOLUCIONAL PROFUNDA”**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Computação Aplicada da Universidade Estadual de Ponta Grossa, pela seguinte banca examinadora:


Prof.^a Dr.^a Maria Salete Marcory Gomes Laz
UEPG


Prof. Dr. Alceu de Souza Britto Junior
UEPG


Dr. Helio Antonio Wood Joris
Fundação ABC

Ponta Grossa, 19 de novembro de 2018.

AGRADECIMENTOS

Agradeço à DEUS, por todas as oportunidades que tive, e por me dar forças para conseguir superar todos os obstáculos, por mais difíceis que fossem.

Agradeço a toda a minha família pelo apoio, em especial meus pais, Ednilson e Marli, que com seu amor e exemplo sempre me motivaram a lutar por meus objetivos e buscar sempre ser uma pessoa melhor.

À minha orientadora, Prof^a. Dr^a. Maria Salete Marcon Gomes Vaz, pela sempre atenciosa orientação durante todo o curso, seus ensinamentos me proporcionaram crescimento intelectual e sua dedicação e conduta no exercício da profissão de docente me servem de expiração.

A todos os professores do programa de Pós-Graduação em Computação Aplicada que tanto contribuíram com o meu amadurecimento como aluno, pesquisador e pessoa, em especial ao Prof^o. Dr^o. Alceu de Souza Britto Junior e Prof^o. Dr^o. Helio Antonio Wood Joris por aceitarem compor a banca e colaborar no aperfeiçoamento deste trabalho.

Aos meus amigos, em especial a Jéssyca Kawana Franquito, pelos tantos anos de amizade, apoio e cafés. Ao Rhandrell de Paula Mainardes, que me auxiliou a entender melhor a problemática tratada neste trabalho. A minha namorada, Jeanine Landmann Senger, que com seu amor e apoio me ajudou em mais essa etapa, e a todos os meus amigos que contribuíram de alguma forma com a minha formação.

*“Ainda que eu fale as línguas dos homens e dos anjos, se não tiver amor,
serei como o sino que ressoa ou como o prato que retine.
Ainda que eu tenha o dom de profecia e saiba todos os mistérios e todo o
conhecimento, e tenha uma fé capaz de mover montanhas, mas não tiver
amor, nada serei.
Ainda que eu dê aos pobres tudo o que possuo e entregue o meu corpo para
ser queimado, mas não tiver amor, nada disso me valerá.
O amor é paciente, o amor é bondoso. Não inveja, não se vangloria, não se
orgulha.
Não maltrata, não procura seus interesses, não se ira facilmente, não guarda
rancor.
O amor não se alegra com a injustiça, mas se alegra com a verdade.
Tudo sofre, tudo crê, tudo espera, tudo suporta.
O amor nunca perece; mas as profecias desaparecerão, as línguas cessarão,
o conhecimento passará.
Pois em parte conhecemos e em parte profetizamos;
quando, porém, vier o que é perfeito, o que é imperfeito desaparecerá.
Quando eu era menino, falava como menino, pensava como menino e
raciocinava como menino. Quando me tornei homem, deixei para trás as
coisas de menino.
Agora, pois, vemos apenas um reflexo obscuro, como em espelho; mas, então,
veremos face a face. Agora conheço em parte; então, conhecerei plenamente,
da mesma forma como sou plenamente conhecido.
Assim, permanecem agora estes três: a fé, a esperança e o amor. O maior
deles, porém, é o amor.”*

(1 Coríntios 13)

RESUMO

As pragas em lavouras causam prejuízos econômicos na agricultura, reduzindo a produção e conseqüentemente os lucros. O manejo de pragas é essencial, para reduzir estes prejuízos, e consiste na identificação e posterior controle desse tipo de ameaça. O controle é fundamentalmente dependente da identificação, pois é a partir dela que o manejo é feito. A identificação é feita visualmente, baseando-se nas características da praga. Essas características são inerentes e diferem de espécie para espécie. Devido à dificuldade da identificação, esse processo é realizado principalmente por profissionais especializados na área, o que acarreta na concentração do conhecimento. Esta dissertação apresenta uma metodologia para classificação de pragas por meio de técnicas de computação, onde um sistema computacional do tipo cliente-servidor foi criado a fim de prover a classificação de pragas por meio de serviço, que é realizado pelo uso de rede neural convolucional baseada na arquitetura Inception V3. As pragas *Anticarsia Gemmatalis*, *Helicoverpa armigera* e *Spodoptera Cosmioides*, foram escolhidas para classificação por serem bastante comuns no estado do Paraná. A rede neural convolucional obteve índice de acerto de 92,5%.

Palavras-chave: Machine Learning; Deep Learning; Rede Neural Convolucional; Classificação de Imagens

ABSTRACT

Pests on crops cause economic damage to agriculture, reducing production and consequently profits. Pest management is essential to reduce these losses, and consists in the identification and subsequent control of this type of threat. Control is fundamentally dependent on identification, because management is done from it. The identification is made visually, based on the characteristics of the pest. These characteristics are inherent and differ from species to species. Due to the difficulty of identification, this process is carried out mainly by professionals specialized in the area, which entails the concentration of knowledge. This dissertation presents a methodology for pest classification by means of computational techniques, in which a client-server computational system was created in order to provide pest classification by means of a service, which is performed by the use of convolutional neural network based in the Inception V3 architecture. The pests *Anticarsia Gemmatalis*, *Helicoverpa armigera* and *Spodoptera Cosmioides*, were chosen for classification because they are quite common in the state of Paraná. The convolutional neural network obtained a success rate of 92.5%.

Keywords: Machine Learning; Deep Learning; Convolutional Neural Network; Images Classification

LISTA DE ILUSTRAÇÕES

Figura 1- Spodoptera cosmioides.	15
Figura 2- Helicoverpa armigera.....	16
Figura 3- Anticarsia gemmatalis.	16
Figura 4- Representação matricial da imagem	19
Figura 5 - Imagem digital da cidade de Songdo, Coréia do Sul.....	20
Figura 6- Relação de PPI.....	21
Figura 7- Representação de DPI.....	21
Figura 8- Representação da diferença entre PPI e DPI	22
Figura 9- (a) Representação em tonalidades de cor do número 8 em escala de cinza. (b) A mesma imagem representada por meio de uma matriz numérica.....	22
Figura 10 – Representação da imagem em um vetor numérico.	23
Figura 11- Esquema cubo de cores do padrão RGB.....	24
Figura 12- Representação do sistema de cores RGB.....	24
Figura 13 - Representação gráfica do padrão HSV	26
Figura 14 - Representação gráfica do padrão HSV II	27
Figura 15- Representação do esquema CIELAB.....	28
Figura 16 - Imagem em tom de cinza	29
Figura 17 - Histograma da imagem em tom de cinza.....	29
Figura 18 - (a) Imagem colorida e (b) seu histograma.	30
Figura 19 - Etapas do Sistema de Visão Artificial	31
Figura 20 - Imagem original (a) e imagem filtrada (b).....	33
Figura 21 - Imagem original (a) e imagem após a aplicação do filtro de mediana (b)...	34
Figura 22- Imagens com ruído.....	34
Figura 23 - Imagens após a aplicação de filtro.....	35
Figura 24- Exemplo de um processo de segmentação: (a) uma imagem original; (b) imagem com aplicação de filtro; e (c) imagem segmentada.	35
Figura 25 - Processo de extração de características	37
Figura 26 - Estágios da seleção de pontos-chave. (a) A imagem original de 233×189 pixels. (b) Os 832 pontos-chave iniciais baseados no máximo e no mínimo da função de diferença de Gauss. Os pontos-chave são exibidos como vetores, indicando escala, orientação e localização. (c) Após aplicar um limite no contraste mínimo, restam 729 pontos-chave. (d) Os 536 pontos chave finais que permanecem após o limiar na relação de curvaturas principais.	38
Figura 27- Imagem com pontos-chave e sua orientação usando SIFT.....	39
Figura 28- A transformada de Laplace distingue círculos brilhantes em fundos escuros da situação inversa.....	40
Figura 29- Convolução com filtro de caixa	41
Figura 30- Imagem com pontos-chave calculados pelo algoritmo BRISK.	42
Figura 31- Classificação de imagens de intensidade variável usando (a) SIFT, (b) SURF e (c) ORB.....	43
Figura 32- Correspondência da imagem original com a mesma imagem rotacionada usando (a) SIFT, (b) SURF e (c) ORB	44
Figura 33- Correspondência da imagem original com a mesma imagem com escala diferente usando (a) SIFT, (b) SURF e (c) ORB.....	45
Figura 34- Correspondência da imagem original com a mesma imagem distorcida usando (a) SIFT, (b) SURF e (c) ORB	46
Figura 35- Correspondência da imagem original com a mesma imagem com efeito “olho de peixe” usando (a) SIFT, (b) SURF e (c) ORB	47

Figura 36 - Correspondência da imagem original com a mesma imagem com efeito “sal e pimenta” usando (a) SIFT, (b) SURF e (c) ORB.....	48
Figura 37- Diferentes técnicas de processamento de imagens	49
Figura 38 - Estrutura geral de uma árvore de decisão	51
Figura 39 - Representação de árvore de decisão	51
Figura 40 - Representação gráfica $p(x)$ versus entropia	53
Figura 41- Rede neural artificial multicamadas	56
Figura 42- Representação de conexão de camadas de rede neural.....	58
Figura 43- Max-Pooling com um filtro de tamanho 2x2 e tamanho do passo (Stride) ..	59
Figura 44 - Extração de características de imagens feita por níveis diferentes de camadas de CNN.	61
Figura 45- (a) Imagem original. (b) Imagem com recortes sobrepostos.	61
Figura 46- Esquema de processamento de cada recorte	62
Figura 47 - Representação da organização dos vetores de saída com os recortes da imagem original.....	63
Figura 48- Representação da redução da matriz original após a aplicação do algoritmo de max pooling.	64
Figura 49- Resumo do processo de classificação por CNN	64
Figura 50 -Outro exemplo de rede neural convolucional profunda.....	65
Figura 51- Comparação entre Tensorflow, Theano e Torch para o conjunto de dados Penn Treebank	67
Figura 52- Esquema de classificação da rede neural profunda	69
Figura 53 - Script de criação de novas imagens a partir de transformações das imagens originais	82
Figura 54 - (a) Imagem original. (b) Imagem após receber as transformações de rotação e aproximação.....	82
Figura 55 - Arquitetura da CNN Inception-V3	83
Figura 56 - Arquitetura de sistema cliente-servidor	85
Figura 57 – (a) Tela inicial do aplicativo. (b) Tela de seleção de origem da imagem a ser classificada. (c) Tela mostrada em caso de nenhuma praga encontrada na imagem.	85
Figura 58 - Tela mostrada em caso de classificação realizada com sucesso	86
Figura 59- Gráfico de assertividade do modelo treinado	87
Figura 60 - Evolução da entropia cruzada	88
Figura 61 - (a) Imagem de <i>Anticarsia gemmatalis</i> que foi classificada erroneamente como <i>Helicoverpa armigera</i> . (b) <i>Helicoverpa armigera</i>	91
Figura 62 - Outro exemplo de classificação errônea. (a) <i>Anticarsia gemmatalis</i> classificada como <i>Helicoverpa armigera</i> . (b) <i>Helicoverpa armigera</i> real	92

LISTA DE TABELAS

Tabela 1 - Valores de cor no formato RGB.....	25
Tabela 2 - Comparação de características entre Tensorflow, Theano, Torch e Caffe....	66
Tabela 3 - Número de imagens coletadas por espécie.....	81
Tabela 4 - Resultados obtidos pela CNN.....	86
Tabela 5 - Matriz de confusão do resultado dos testes	89
Tabela 6 - Ranking de assertividade por espécie.....	89
Tabela 7 - Porcentagem de classificação incorreta por número de imagens por espécie	90
Tabela 8 - Porcentagem de classificação correta por número de imagens por espécie ..	90

LISTA DE SIGLAS

ABIS	Automated Identification of Bee Species
ACP	Análise de Componentes Principais
AGAST	Adaptative and Generic Accelerated Segment Test
ANN	Artificial Neural Network
ANVISA	Agencia Nacional de Vigilância Sanitária
API	Application Programming Interface
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
CNN	Convolutional Neural Network
CNTK	Cognitive Toolkit
CPU	Central Process Unit
CSS	Cascading Style Sheets
DAISY	Digital Automated Identification System
DDG	Diferença de Gaussianas
DPI	Dots Per Inch
EF	Elíptica de Fourier
FREAK	Fast Retina Keypoint
GPU	Graphics Processing Unit
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
JPEG	Joint Photographic Experts Group
MLP	Multilayer Perceptron
NNC	Nearest-neighbour Classification
ORB	Oriented FAST and Rotated BRIEF
PIB	Produto Interno Bruto
PPI	Pixels Per Inch
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features

SPIDA	Species Identified Automatically
SVA	Sistema de Visão Artificial
SVM	Support Vector Machine

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVO GERAL	17
1.2	OBJETIVOS ESPECÍFICOS	18
1.3	ESTRUTURA	18
2	REVISÃO DE LITERATURA	18
2.1	MÉTODOS DE CLASSIFICAÇÃO DE PRAGAS	18
2.2	IMAGENS DIGITAIS	19
2.2.1	ESPAÇO DE CORES	22
2.2.2	HISTOGRAMA.....	28
2.3	MÉTODOS COMPUTACIONAIS	30
2.3.1	ESTRUTURA DE UM SISTEMA DE VISÃO ARTIFICIAL	31
2.3.2	AQUISIÇÃO DE IMAGENS	31
2.3.3	BASE DE CONHECIMENTO	32
2.3.4	PRÉ-PROCESSAMENTO	32
2.3.5	SEGMENTAÇÃO	35
2.3.6	EXTRAÇÃO DE CARACTERÍSTICAS	36
2.3.7	CLASSIFICAÇÃO	49
2.3.7.1	ÁRVORE DE DECISÃO	50
2.4	MACHINE LEARNING	54
2.5	DEEP LEARNING	54
2.6	REDE NEURAL CONVOLUCIONAL	57
2.7	TENSORFLOW.....	65
3	TRABALHOS CORRELATOS	68
4	METODOLOGIA	80
4.1	AQUISIÇÃO DE IMAGENS	81

4.2	SISTEMA COMPUTACIONAL.....	83
5	RESULTADOS E DISCUSSÃO.....	86
6	CONCLUSÕES E TRABALHOS FUTUROS.....	92
7	REFERÊNCIAS.....	94

1 INTRODUÇÃO

As pragas são responsáveis por perda de produtividade e conseqüentemente geram prejuízos no setor agrônômico. Estudos realizados em 2015 (AGROLINK, 2015), mostram que cerca de 7,7% da produção média anual brasileira é perdida por causa de ataques de lagartas, moscas e outras doenças, totalizando em torno de 25 milhões de toneladas, significando uma perda anual de até R\$ 55 bilhões. No âmbito mundial, as perdas causadas por pragas chegaram a cerca de 5% do Produto Interno Bruto (PIB) mundial, correspondendo aproximadamente US\$ 1,4 trilhão.

O combate às pragas é feito por meio de inseticidas, produtos químicos amplamente usados em lavouras, cujo uso atinge cerca de 70% dos alimentos produzidos no Brasil, segundo dados divulgados pela Agência Nacional de Vigilância Sanitária (ANVISA) (EL PAÍS, 2015).

Como parte do processo de combate às pragas, é necessário identificá-las, para que o tratamento adequado seja realizado. É importante frisar que a identificação incorreta da praga pode ocasionar no tratamento inadequado da mesma, com ineficiência do processo e desperdício de recursos, bem como o uso desnecessário de produtos químicos, causando um impacto desnecessário ao meio ambiente.

É de vital importância a identificação correta de pragas, seja na pesquisa entomológica ou no controle de pragas propriamente dito. No entanto, a identificação de espécies, usando métodos tradicionais é um processo demorado, devido ao grande número de espécies de insetos que deve ser identificado e os recursos taxonômicos são limitados.

O termo “reconhecimento” tem sido usado para se referir a muitos aspectos visuais, incluindo identificação (fisicamente igual), categorização (atribuição de um objeto a uma categoria) e discriminação (atribuição de um objeto a uma classe). Normalmente, quando falamos em reconhecer um objeto, queremos dizer que categorizamos com sucesso uma instância de uma determinada classe de objeto. (LITER; BÜLTHOFF, 1998)

A identificação rotineira de espécies tem muitas peculiaridades e as atividades podem ser automatizadas, representando as rotinas desempenhadas por profissionais da biologia pura e aplicada.

A classificação de pragas é realizada, geralmente, por profissionais especializados na área, acarretando na concentração do conhecimento e assim gerando

dependência desses profissionais. Além disso, muitos desses profissionais estão atrelados a empresas e podem indicar produtos químicos das empresas das quais representam. Sendo assim, o agricultor, por exemplo, pode não ter outras opções de produtos a serem usados, produtos esses que poderiam ter um custo menor e ter o mesmo grau de efetividade no tratamento.

Devido à importância econômica que a cultura da Soja representa para o estado do Paraná, podemos destacar as pragas mais comuns que a atacam, destacam-se *Spodoptera cosmioides*, *Helicoverpa armigera* e *Anticarsia gemmatalis*. A *Spodoptera cosmioides*, também conhecida como “Lagarta das Vagens” ou “Lagarta-marrom”, danifica as vagens e grãos da planta, também realiza perfuração dos botões florais e dos frutos. Possui ovos de cor esverdeada-clara. As lagartas inicialmente são marrons, em fase adulta podem chegar a 4 cm de comprimento e se tornar cinzas, castanhas ou pretas. No dorso, possui linhas longitudinais de cor branca, marrom ou alaranjada, com pontuações douradas ou brancas.

Figura 1- *Spodoptera cosmioides*.



Fonte: Agência Embrapa de Informação Tecnológica. Disponível em: <<http://www.agencia.cnptia.embrapa.br/gestor/soja/arvore/CONT000fznzu9ib02wx5ok0cpoo6achc3ixw.html>>. Acesso em: 11 de set. 2017

A *Helicoverpa armigera* possui ovos de 0,5 mm de diâmetro e mudam de branco para marrom para cabeça preta antes da incubação. Larvas recém-incubadas são de cor clara com pequenas manchas e cabeça escuras. À medida que as larvas se desenvolvem, elas ficam mais escuras e os pontos mais escuros se tornam mais visíveis. Larvas médias

desenvolvem linhas que correm no comprimento do corpo e são de cor variável, as grandes larvas de *Helicoverpa armigera* têm pelos brancos ao redor da cabeça.

Figura 2- *Helicoverpa armigera*.



Fonte: Rural Pecuária. Disponível em: <<http://ruralpecuaria.com.br/tecnologia-e-manejo/pragas/helicoverpa-armigera-qual-a-situacao-atual-da-praga.html>>. Acesso em: 11 de set. 2017.

As lagartas da espécie *Anticarsia gemmatalis* são pretas ou verdes, com pequenas listras no dorso e nas laterais do corpo. Quando se sentem ameaçadas, as larvas desta espécie expelem uma secreção de cor acastanhada, que forma um fio, por meio do qual elas saltam, dependuram-se e retorcem-se no ar.

Figura 3- *Anticarsia gemmatalis*.



Fonte: Agência Universitária de Notícias – Universidade de São Paulo. Disponível em: < <http://www.usp.br/aun/antigo/exibir?id=6575&ed=1161&f=2> >. Acesso em: 11 de set. 2017.

Ao avaliarmos esses aspectos, podemos verificar a possibilidade de aplicação de um método computacional para ajudar os produtores rurais ou qualquer outra pessoa que tenha interesse na classificação de pragas. Por meio de uma solução computacional, onde as técnicas de processamento de imagens possam ser empregadas a fim de identificar as pragas.

1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é especificar a classificação de pragas, para tanto, foi utilizada como base as espécies *Anticarsia Gemmatalis*, *Helicoverpa armigera* e *Spodoptera Cosmioides* na fase de larva. Essas lagartas foram escolhidas por serem bastante comuns no estado do Paraná. A classificação de pragas foi realizada por meio de rede neural convolucional profunda, que foi treinada em uma base de imagens que contém 750 imagens das pragas citadas e o índice de assertividade foi medido por meio de uma base de imagens contendo 250 imagens que não foram usadas no treinamento da rede. A classificação realizada pela rede neural convolucional foi publicada como serviço em um sistema computacional do tipo cliente-servidor, cujo objetivo é atender as requisições das aplicações “cliente”. O serviço exige um padrão simples de entrada de dados, fazendo

com que seja facilmente utilizado por qualquer tipo de “cliente”, contanto que o mesmo respeite a forma com que o serviço deve ser utilizado.

1.2 OBJETIVOS ESPECÍFICOS

- Criação de base de imagens de pragas, que será utilizada no treinamento e validação da rede neural convolucional;
- Utilização de arquitetura de rede neural convolucional aplicada na classificação das pragas.
- Criação de um método computacional, do qual será composto por arquitetura do tipo cliente-servidor, onde o cliente irá enviar as imagens que deseja classificar para o servidor, e o servidor por sua vez, envia o resultado do processamento para o cliente.

1.3 ESTRUTURA

Este trabalho está estruturado em 6 capítulos, onde o primeiro capítulo trata da introdução. No capítulo de número 2 são apresentados os referenciais teóricos que embasam o presente trabalho, neste capítulo são tratados os conceitos e técnicas de processamento digital de imagens bem como a revisão bibliográfica e no capítulo 3 são apresentados os trabalhos correlatos. No capítulo 4 são descritos a metodologia e materiais utilizados no desenvolvimento deste trabalho. No capítulo 5 os resultados são apresentados e discutidos, seguido das conclusões e perspectiva de trabalhos futuros que serão abordadas no capítulo 6.

2 REVISÃO DE LITERATURA

2.1 MÉTODOS DE CLASSIFICAÇÃO DE PRAGAS

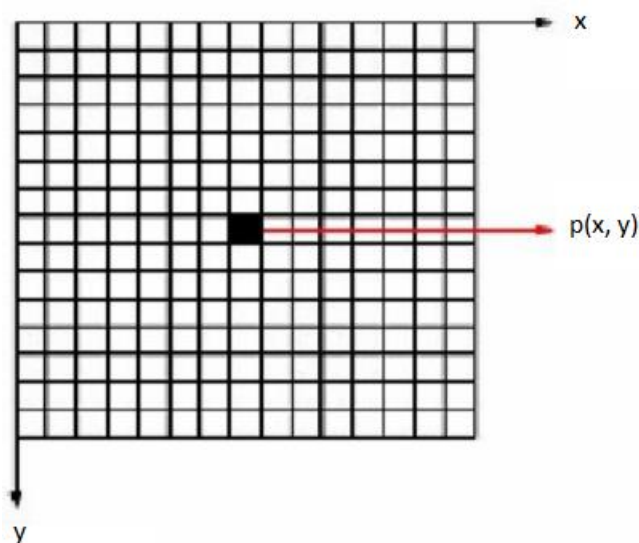
A classificação de pragas faz parte do conjunto da ciência de classificação de seres vivos chamada Taxonomia, essa ciência busca descrever, identificar e classificar os seres vivos, a classificação permite organizar os conhecimentos. No entanto, para que seja possível classificar, é necessário estabelecer critérios, ou seja, analisar semelhanças e diferenças presentes entre os objetos que se deseja separar. Dessa forma, a Taxonomia baseia-se nas características físicas dos seres vivos para classifica-los.

2.2 IMAGENS DIGITAIS

Computadores não interpretam imagens da mesma forma que os seres vivos, sendo assim, se faz necessário um modo para que seja possível para essas máquinas poderem interpretar as imagens, com isso, as imagens são convertidas em um conjunto numérico.

A definição de imagem digital pode ser compreendida por meio da função bidimensional da luz $f(x, y)$, onde x e y são as representações de coordenadas espaciais, e seu valor é relativo ao brilho ou nível de cinza (GONZALEZ; WOODS, 2010). Já a imagem digital, é compreendida toda imagem $f(x, y)$ discretizada, em coordenadas espaciais e brilho e seus índices de linhas e colunas indicam uma localização na imagem e seu valor representa o nível de cinza naquele ponto. As imagens digitais podem ser representadas como uma matriz de duas dimensões.

Figura 4- Representação matricial da imagem



Fonte: O autor.

Os elementos da matriz são chamados de pixels, resultado da junção das palavras em inglês “*Picture*” e “*Element*”, ou seja, elemento de imagem. Um pixel é o menor ponto que forma uma imagem digital, sendo que o conjunto de pixels formam a imagem inteira.

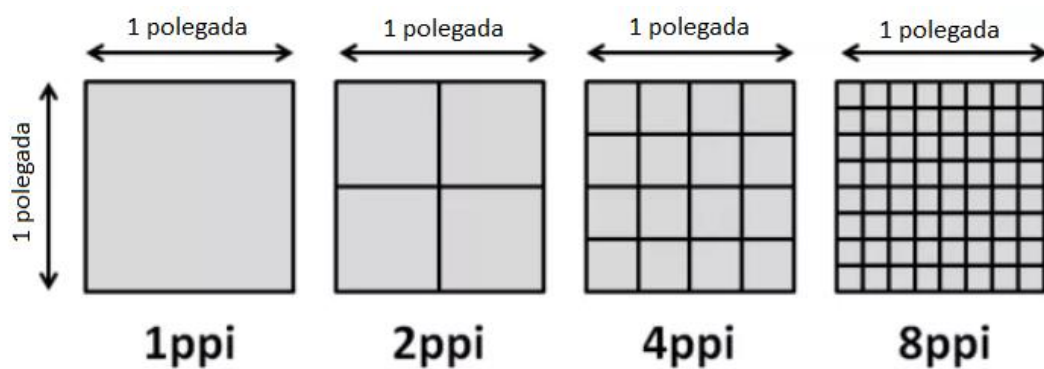
Figura 5 - Imagem digital da cidade de Songdo, Coréia do Sul.



Fonte: Business Insider. Disponível em: <<https://www.businessinsider.com/songdo-south-korea-design-2017-11>>. Acesso em: 20 de set. 2017.

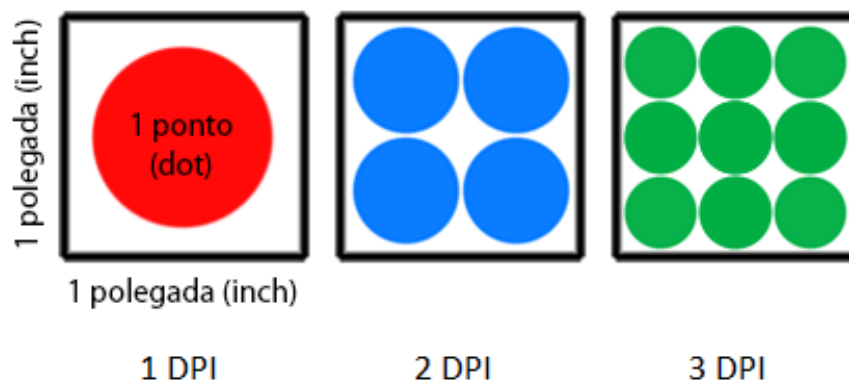
A densidade de pixels contidos na imagem reflete-se na nitidez da mesma, quanto mais densa a imagem melhor será a sua definição e detalhamento. A densidade de pixels é comumente chamada de resolução. Podemos calcular a resolução da imagem por meio do chamado DPI (pontos por polegada ou do inglês “*Dots Per Inch*”) e PPI (pixels por polegada ou “*Pixels Per Inch*”). O DPI é utilizado para representar o valor da quantidade e espaçamento entre pontos magentas, cianos, amarelos e pretos em imagens que foram reproduzidas em algum tipo de material, já o PPI representa o valor de pixels na tela. O DPI representa o número de pontos que podem ser encontrados em uma polegada (2.54 cm) de uma determinada imagem. Os pontos de uma imagem apresentada em um monitor de vídeo consistem no PPI, mas fabricantes de equipamentos e utilizadores utilizam a designação DPI tanto para imagens impressas quanto em monitores.

Figura 6- Relação de PPI



Fonte: Adaptado de “A diferença entre PPI e DPI”. Disponível em <<https://designculture.com.br/a-diferenca-entre-ppi-e-dpi>>. Acesso em 25 de mar. 2018.

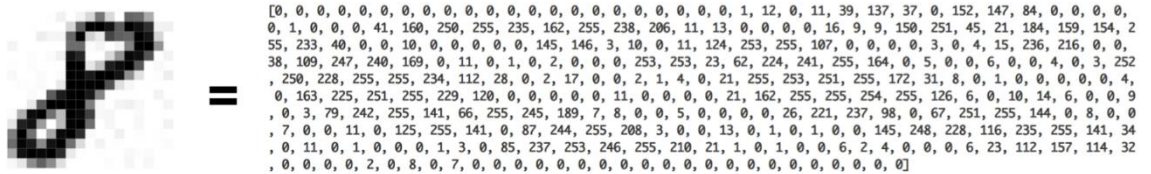
Figura 7- Representação de DPI



Fonte: O autor.

Fonte: Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Disponível em <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>. Acesso em: 25 de mar. 2018.

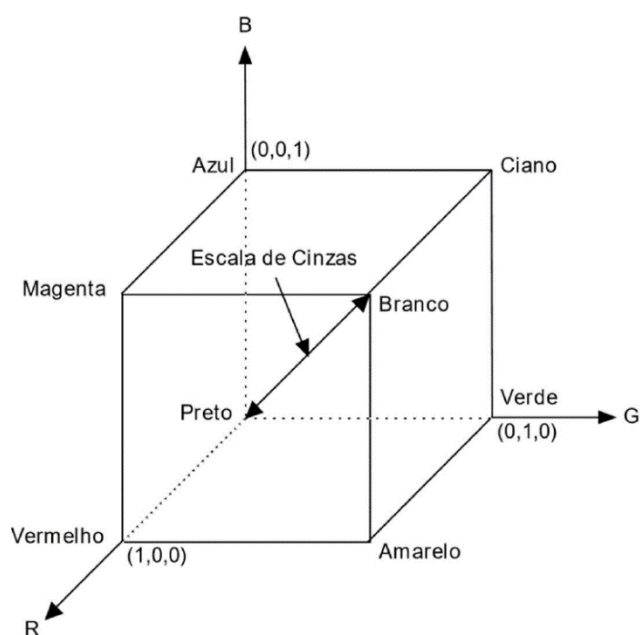
Figura 10 – Representação da imagem em um vetor numérico.



Fonte: Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Disponível em <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>. Acesso em: 25 de mar. 2018.

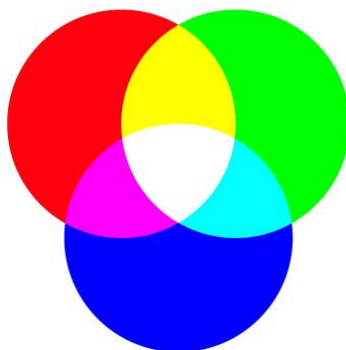
Contudo, nas imagens coloridas, cada pixel é composto por um conjunto de 3 valores: vermelho, verde e azul. Essa representação é chamada de RGB (do Inglês *Red, Green, Blue*) e nela os valores de vermelho, verde e azul são combinados de várias formas de modo a reproduzir um largo espectro cromático. O modelo de cores RGB foi desenvolvido baseado na teoria de visão colorida tricromática, de Young-Helmholtz, e no triângulo de cores de Maxwell.

Figura 11- Esquema cubo de cores do padrão RGB



Fonte: (MARQUES FILHO; VIEIRA NETO, 1999).

Figura 12- Representação do sistema de cores RGB



Fonte: o autor.

Por meio das diversas combinações entre as cores vermelho, verde e azul, muitas outras cores podem ser representadas dessa forma. Por exemplo um monitor que utiliza até 24 bits de informação para cada pixel, e onde a divisão é de 8 bits para cada uma das cores (vermelho, verde e azul), possibilita assim um alcance de 256 possíveis valores para

cada cor. Dessa forma, obtemos mais de 16 milhões (16.777.216 ou 256^3) de diferentes combinações de tons, saturação e brilho.

Na representação de cores onde a escala utilizada varia de 0 a 255, obtemos os seguintes valores:

Tabela 1 - Valores de cor no formato RGB

Cor	Valor (RGB)
Branco	255, 255, 255
Azul	0, 0, 255
Vermelho	255, 0, 0
Verde	0, 255, 0
Amarelo	255, 255, 0
Magenta	255, 0, 255
Ciano	0, 255, 255
Preto	0, 0, 0

Fonte: O autor.

O padrão RGB possui um limite de representação de cores, podemos perceber que o menor valor possível RGB (0,0,0) representa a cor Preta e o maior valor possível RGB (255, 255, 255) representa a cor Branca, e com isso as demais cores variam dentro deste limite. Existem outras formas de representação de imagens digitais por meio de cores, além do sistema RGB, podemos citar HSV, e CIE L*a*b*, vejamos cada um deles:

HSV é um sistema de cores formado pelos componentes *hue*, *saturation* e *matiz* (do Inglês *matiz*, saturação e valor). O HSV também é conhecido como HSB (*hue*, *saturation* e *brightness* — *matiz*, saturação e brilho, respectivamente).

Matiz (tonalidade): Verifica o tipo de cor, abrangendo todas as cores do espectro. Seus limites são de 0 a 360, mas para algumas aplicações, esse valor é normalizado de 0 a 100%.

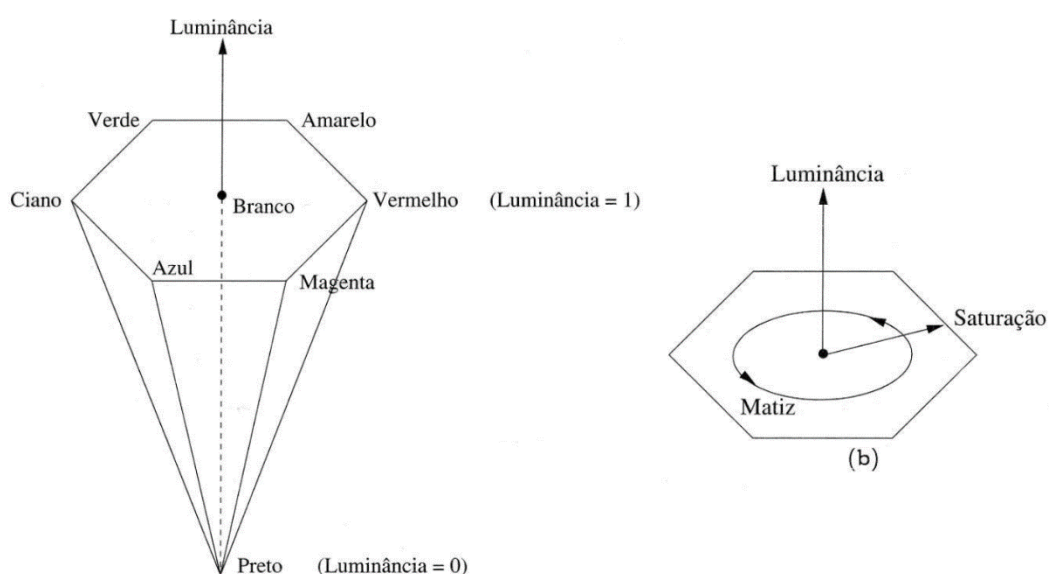
Saturação: Quantidade de luz branca presente. Quanto menor esse valor, mais com tom de cinza aparecerá a imagem. Quanto maior o valor, mais "pura" é a imagem. Atinge valores de 0 a 100%.

Valor (brilho): Define o brilho da cor. Atinge valores de 0 a 100%.

Esse sistema foi inventado no ano de 1974, por Alvy Ray Smith. É caracterizada por ser uma transformação não-linear do sistema de cores RGB, e ao contrário do RGB, o modelo HSV não tem a divisão de canais por cores, mas sim pelos componentes que o compõe.

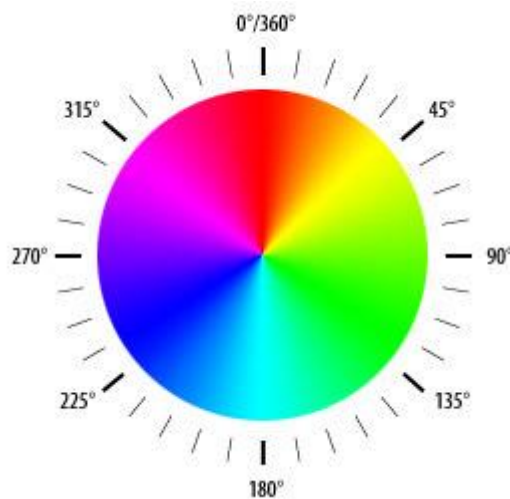
A separação das informações de cor da intensidade torna o espaço de cor HSV uma ferramenta ideal para o desenvolvimento de algoritmos de processamento digital de imagens. Geometricamente, o espaço de cor HSV pode ser visto como uma pirâmide hexagonal derivada do cubo RGB (PEDRINI; SCHWARTZ, 2008).

Figura 13 - Representação gráfica do padrão HSV



Fonte: (PEDRINI; SCHWARTZ, 2008).

Figura 14 - Representação gráfica do padrão HSV II



Fonte: Public Lab. Disponível em: <<https://publiclab.org/notes/warren/08-24-2013/infragrammar-with-hsv-color-model>>. Acesso em 13 de set. 2017.

O CIE $L^*a^*b^*$ (CIELAB) é um espaço de cor especificado pela Comissão Internacional de Iluminação (Comissão Francesa Internacional de l'Clairage, de onde surgiu a sua inicialização CIE). Esse espaço de cor descreve todas as cores visíveis ao olho humano e foi criado para servir como um modelo independente de dispositivo para ser usado como referência.

As três coordenadas do CIELAB representam a luminosidade da cor, onde $L^* = 0$ é preto e $L^* = 100$ indica branco difuso; branco especular pode ser maior), sua posição entre vermelho/magenta e verde.

O valor a^* representa valores que quando negativos, indicam verde. Valores positivos indicam magenta, e sua posição entre amarelo e azul.

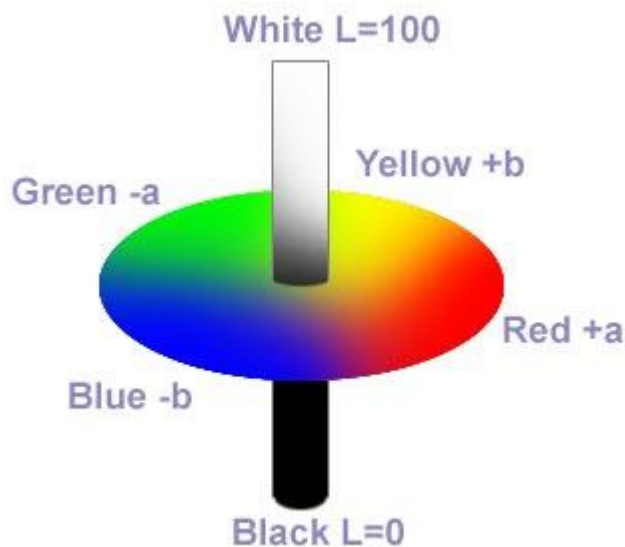
O valor b^* representa valores que quando negativos indicam valores azuis. Valores positivos indicam amarelo. O asterisco (*) depois de L, a e b são pronunciados “estrela” e fazem parte do nome do sistema.

Como o modelo $L^* a^* b^*$ é um representado por uma matriz tridimensional, ele pode ser representado apenas em um espaço tridimensional. Vale ressaltar que as representações visuais da gama completa de cores neste modelo não são precisas; e existem apenas para ajudar na compreensão do conceito.

O CIELAB é um espaço de cores de valor cromático, já que os canais oponentes vermelho-verde e amarelo-azul são computados como diferenças de transformações de leveza das respostas do cone (putativo). Não possui direitos autorais ou licença, como é

totalmente definido matematicamente, o modelo CIELAB é de domínio público, e por essa razão é usado sem restrições.

Figura 15- Representação do esquema CIELAB



Fonte: PCE Instruments. Disponível em: https://www.google.com.br/url?sa=i&source=images&cd=&ved=2ahUKEwjngYOk4d7cAhWMCpAKHVRAhEQjRx6BAgBEAU&url=https%3A%2F%2Fwww.pce-instruments.com%2Fenglish%2Fmeasuring-instruments%2Ftest-meters%2Fcolor-meter-colour-meter-kat_40051_1.htm&psig=AOvVaw1q1WAsnl3BZJL2-YG4Kg28&ust=1533862502334604. Acesso em 11 de set. 2017.

2.2.2 HISTOGRAMA

O histograma de uma imagem é simplesmente um conjunto de números indicando o percentual de pixels naquela imagem que apresentam um determinado nível de cinza. Estes valores são normalmente representados por um gráfico de barras que fornece para cada nível de cinza o número (ou o percentual) de pixels correspondentes na imagem. Através da visualização do histograma de uma imagem obtemos uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura). (MARQUES FILHO; NETO, 1999).

O histograma é calculado a partir da equação:

$$p_r(r_k) = \frac{n_k}{n}$$

onde:

$$0 \leq r_k \leq 1$$

$k = 0, 1, \dots, L-1$, onde L representa a quantidade de níveis de cinza;

n = número total de pixels na imagem;

$p_r(r_k)$ = probabilidade do k-ésimo nível de cinza;

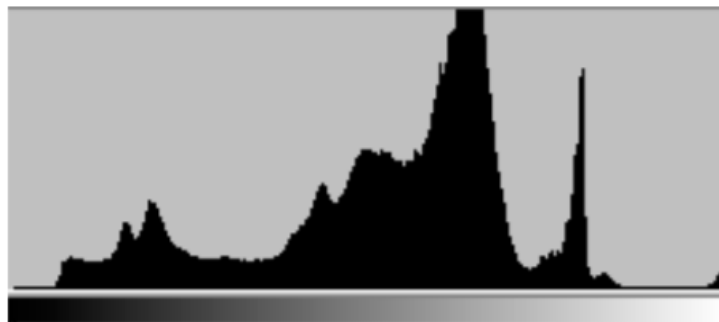
n_k = número de pixels cujo nível de cinza corresponde a k.

Figura 16 - Imagem em tom de cinza



Fonte: (CONCI, 2015)

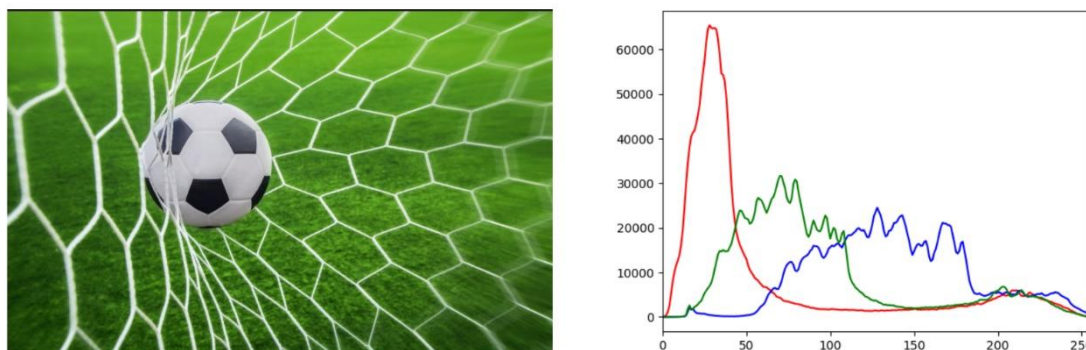
Figura 17 - Histograma da imagem em tom de cinza.



Fonte: (CONCI, 2015).

Imagens escuras tendem a concentrar maiores quantidades ou percentuais de pixels, próximos ao 0 (zero), enquanto imagens claras tem maior número de pixels próximo ao 255 (MARQUES FILHO & VIEIRA NETO, 1999).

Figura 18 - (a) Imagem colorida e (b) seu histograma.



Fonte: O autor.

Nota: 1: Na imagem (b), podemos observar a distribuição dos pixels baseados no sistema de cor RGB. Onde no eixo x temos a frequência e no eixo y temos o valor do pixel.

2.3 MÉTODOS COMPUTACIONAIS

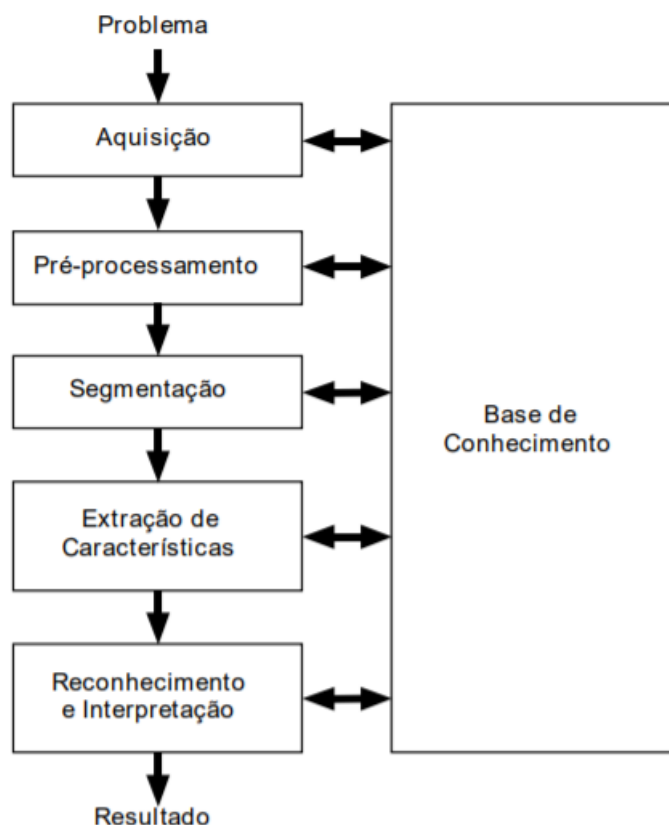
O campo de processamento digital de imagens (PDI) consiste em desenvolver técnicas de compreensão dos mecanismos de visão natural e artificial e em fornecer ferramentas cujo desempenho aproxima-se o mais possível das extraordinárias capacidades do sistema visual humano (FACON, 2002). O processamento de imagens é uma área multidisciplinar que envolve diversas áreas como: Eletrônica, Ótica, Inteligência Artificial, Ciência da Computação e outras. O uso do processamento digital de imagens teve seu início na década de 1920, tendo maior avanço na década de 1960. (MARQUES FILHO; VIEIRA NETO, 1999).

2.3.1 ESTRUTURA DE UM SISTEMA DE VISÃO ARTIFICIAL

Um sistema de visão artificial (SVA) é sistema computadorizado capaz de adquirir, processar e interpretar imagens correspondentes a cenas reais. (MARQUES FILHO; VIEIRA NETO, 1999).

Podemos descrever o SVA a partir do seguinte diagrama:

Figura 19 - Etapas do Sistema de Visão Artificial



Fonte: (MARQUES FILHO; VIEIRA NETO, 1999).

Vale ressaltar que as características do SVA podem variar conforme o problema a ser solucionado, ou seja, algumas dessas etapas poderão não ser necessárias. Vejamos a descrição detalhada de cada uma das etapas citadas acima.

2.3.2 AQUISIÇÃO DE IMAGENS

Ao iniciarmos o processo, devemos traçar qual a melhor estratégia para obtenção das imagens que serão utilizadas a fim de solucionar a problemática em questão. Para tanto, é necessário compreender a fundo o problema e então realizar a aquisição das

imagens que melhor poderão ajudar na resolução. Tomemos como exemplo a problemática da classificação de pragas, onde devemos primeiramente propor quais os tipos de pragas que iremos classificar. No presente trabalho, foram escolhidas espécies de pragas comuns no estado do Paraná. Uma vez delimitado o escopo da classificação, podemos realizar a aquisição das imagens das pragas em questão.

2.3.3 BASE DE CONHECIMENTO

A base de conhecimento tem tamanho e complexidade conforme o problema a ser solucionado. A base de conhecimento serve como guia para o funcionamento de cada etapa e também deve permitir a realimentação entre elas. A realimentação é importante pois permite agregar continuamente novas imagens, contribuindo assim para a melhoria do SVA. A grosso modo, quanto mais imagens tivermos em nossa base de conhecimento, a tendência é de obtermos melhores resultados no que tange a tarefa de classificação, pois partimos do princípio de que quanto mais imagens coerentes à solução proposta, maior é a base de conhecimento que utilizaremos para extrair as características dos objetos presentes nessas imagens e por fim mais informações dos descritores para classificar posteriormente.

2.3.4 PRÉ-PROCESSAMENTO

Após a aquisição das imagens ter sido realizada, algumas imagens podem apresentar imperfeições, e então o pré-processamento é utilizado a fim de solucionar ou amenizar problemas de ruído, contraste ou brilho inadequado, entre outras. O ruído não é apenas uma interferência no sinal de captura da imagem, pode ser também interferências que prejudicam a interpretação ou o reconhecimento de objetos na imagem (MARENGONI & STRINGHINI, 2009).

A função da etapa de pré-processamento é reduzir ou eliminar as imperfeições da imagem. A manipulação dos valores de intensidade do pixel é chamada de operações de baixo nível. A etapa de pré-processamento é bastante importante porque é por meio dela que obtemos imagens de maior qualidade do que a original.

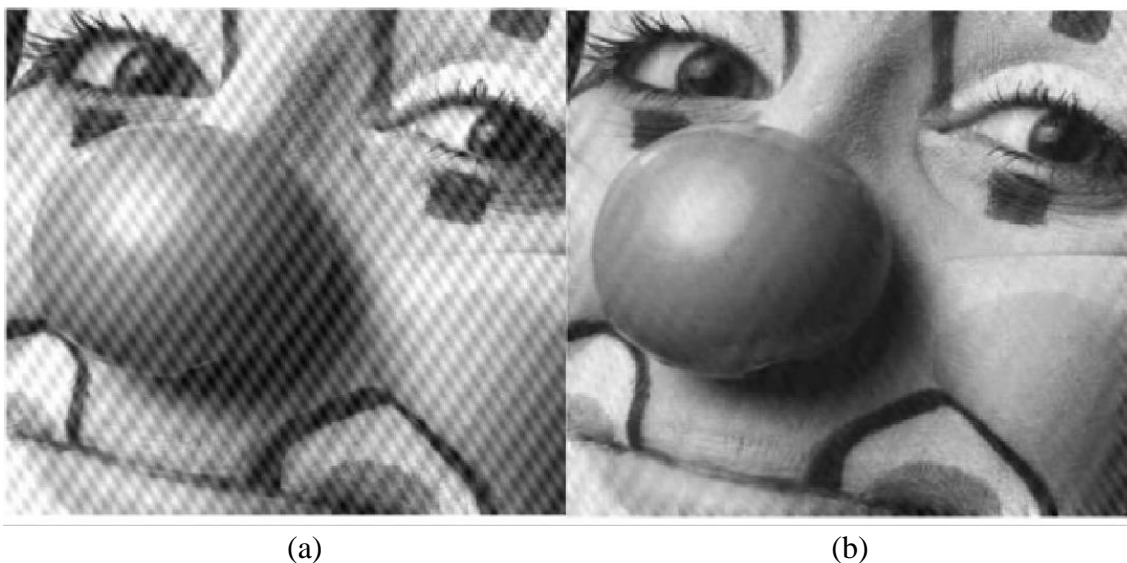
Uma das ferramentas mais utilizadas no pré-processamento para remoção de ruídos são os filtros. Os chamados filtros espaciais atuam diretamente na imagem, e essas funções são expressas pela seguinte equação:

$$g(x, y) = T[f(x, y)]$$

onde:

- $g(x, y)$ é a imagem processada.
- $f(x, y)$ é a imagem original,
- T é um operador em f , definido em uma certa vizinhança de (x, y) .

Figura 20 - Imagem original (a) e imagem filtrada (b).

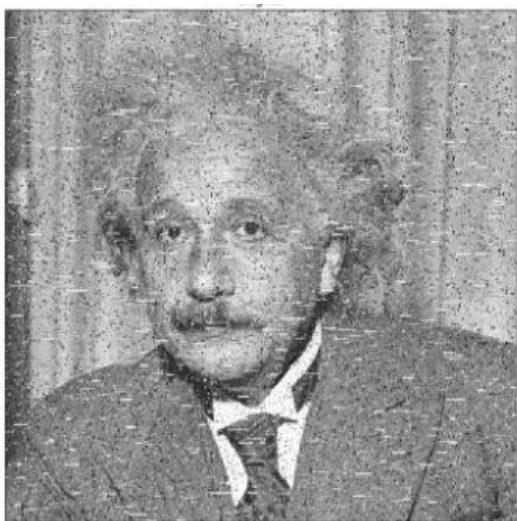


Fonte: (MARENGONI & STRINGHINI, 2009).

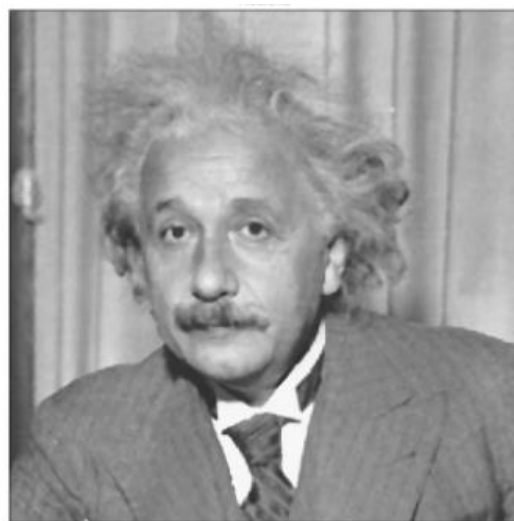
Nota: 1: Podemos perceber que as linhas diagonais na figura (a) tornaram-se suavizadas na figura (b) devido ao processo de filtro.

A suavização de imagens no domínio espacial baseia-se no uso de máscaras de convolução adequadas para o objetivo em questão, normalmente detalhes que não são de interesse são eliminados para as etapas subseqüentes do processamento, ou a remoção de ruídos nela presentes. Dentre as técnicas mais conhecidas de suavização estão a filtragem pela média e o filtro da mediana. (GONZALEZ; WOODS, 2010).

Figura 21 - Imagem original (a) e imagem após a aplicação do filtro de mediana (b).



(a)

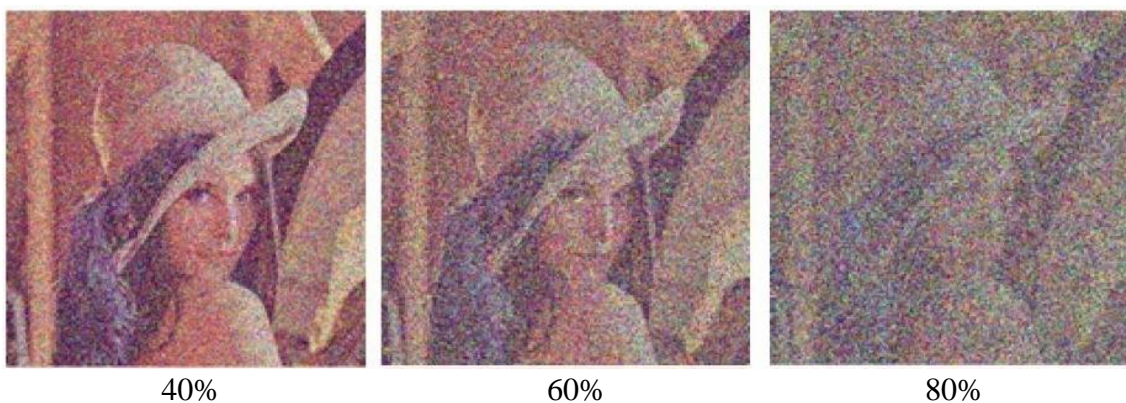


(b)

Fonte: (GONZALEZ; WOODS, 2010).

No filtro de mediana os valores dos pixels contidos em uma determinada região da imagem (máscara) são ordenados e o valor que ocupa a posição mediana é selecionado para a posição (x, y) da imagem filtrada. Esse método reduz o efeito de ruído de pulso conhecido como “sal e pimenta” uma vez que valores pontuais raramente aparecem juntos e, portanto, nunca ocupam a posição mediana (MARENGONI & STRINGHINI, 2009).

Figura 22- Imagens com ruído



Fonte: (IRUM, Isma et al, 2015).

Figura 23 - Imagens após a aplicação de filtro

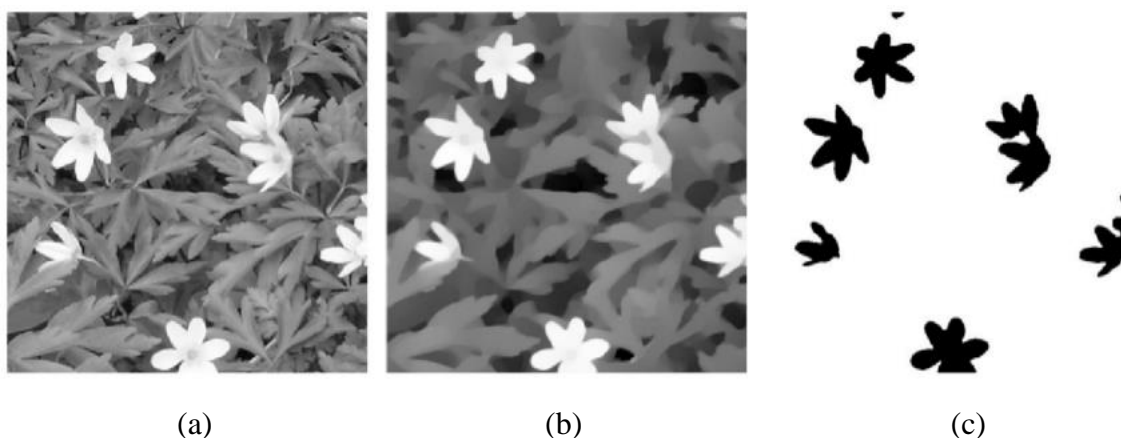


Fonte: (IRUM, Isma et al, 2015).

2.3.5 SEGMENTAÇÃO

É a denominação dada a etapa de dividir uma imagem em unidades de interesse. Embora seja considerada uma das tarefas mais difíceis, tem grande influência no resultado final, pois quanto mais precisa for a segmentação, maiores as chances de sucesso no reconhecimento dos objetos (GONZALEZ; WOODS, 2010). Existem diversas técnicas de segmentação, mas a maioria se divide entre abordagens de descontinuidade, como detecção de bordas, e de similaridade, como limiarização.

Figura 24- Exemplo de um processo de segmentação: (a) uma imagem original; (b) imagem com aplicação de filtro; e (c) imagem segmentada.



Fonte: (SOLEM, 2012).

Essas unidades de interesse podem ser frontais, de fundo ou um objeto específico a ser detectado na imagem e são definidas com o auxílio de algumas características, como cor, borda ou similaridade (SOLEM, 2012). Ao segmentarmos a imagem onde definimos como critério principal a sua cor, sua eficiência está relacionada ao fato da cor do objeto de interesse ser substancialmente diferente da cor de fundo (DEMAAGD et al., 2012). É possível utilizar um cálculo de distância para fazer a segmentação, como a euclidiana, correspondendo a distância medida através da raiz quadrada da soma dos quadrados das diferenças entre as coordenadas dos pontos em questão (MARQUES FILHO & VIEIRA NETO, 1999).

$$D_e(A, B) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

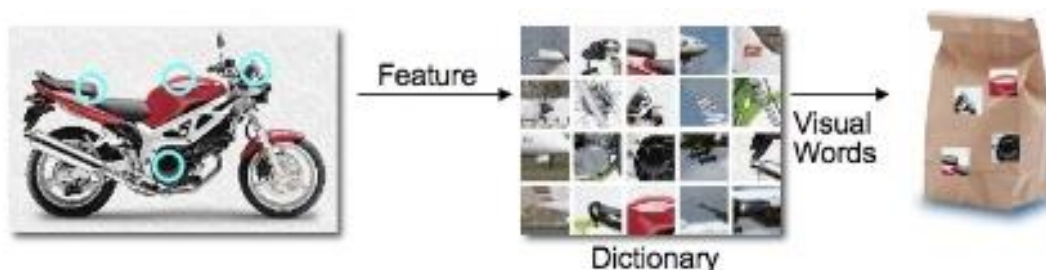
onde:

- $D_e(A, B)$ representam o cálculo da distância euclidiana dos pontos A e B.
- x_a e y_a representam as coordenadas do pixel A.
- x_b e y_b representam as coordenadas do pixel B.

2.3.6 EXTRAÇÃO DE CARACTERÍSTICAS

A etapa de extração de características realiza a denominada função, posterior a etapa de segmentação, por meio de descritores que caracterizam cada informação e que apresentem perceptível distinção. Os descritores são representados por uma estrutura de dados conveniente ao algoritmo de classificação. Vale ressaltar que nesta etapa a entrada ainda é uma imagem, e a saída é um conjunto de dados correspondentes àquela imagem, ou seja, para descrevermos por exemplo um caractere, seus descritores serão as coordenadas x e y de seu centro de gravidade e a razão entre largura e altura, gerando assim um vetor de três elementos, que se encaixa adequadamente ao armazenamento dessas informações.

Figura 25 - Processo de extração de características



Fonte: (SIVIC; ZISSERMAN, 2003).

Nota: 1: Na imagem acima, algumas características são extraídas e então um “dicionário” é criado a partir delas. Esse dicionário será utilizado como um vetor de características.

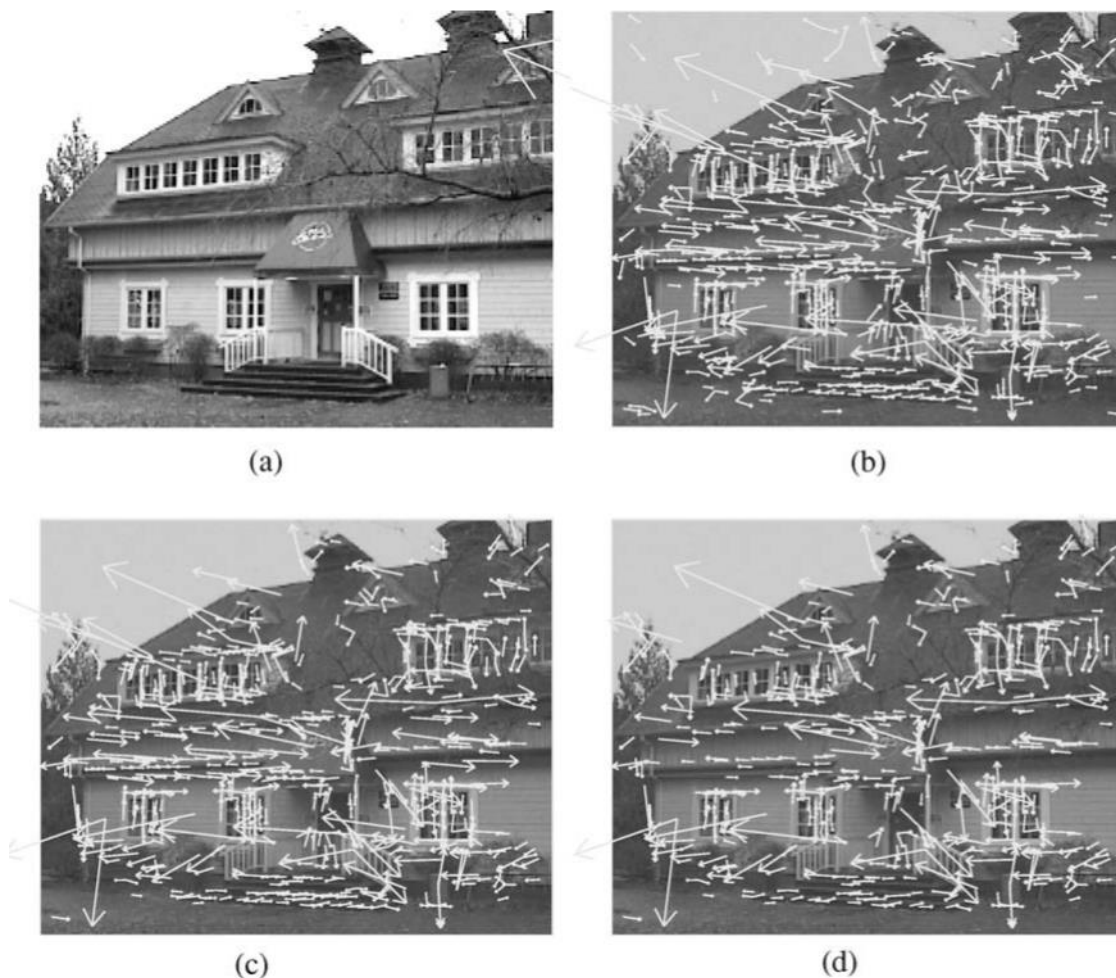
A extração de características transcreve a informação da imagem em pontos de interesse buscando localizar características que sejam relevantes para a tarefa de classificação. A extração de características e classificação de imagem tem sido dois problemas importantes no domínio computacional de visão de máquina e robótica, e suas aplicações continuam a crescer em vários outros campos. Uma técnica de extração de características deve ser robusta para transformações de imagem, como rotação, escala, iluminação, ruído e afim. Além disso, as características extraídas devem ser altamente distintas, de modo que sejam categorizadas com alta probabilidade (MOGHADDAM et al., 2001).

A descrição automática de características é realizada por meio de algoritmos descritores, vejamos alguns deles:

Scale-Invariant Feature Transform (SIFT): Amplamente usado, o SIFT é tanto descritor quanto detector de características, é baseado em Diferenças de Gaussianas. Como descritor, ele calcula o histograma do gradiente da imagem. O descritor resultante é um vetor de características de 128 posições.

SIFT foi proposto por Lowe (LOWE, 2004) a fim de trabalhar com a rotação, transformações, intensidade e mudança de ponto de vista na correspondência de características. O algoritmo SIFT realizada 4 etapas básicas, onde estima um espaço de escala extrema usando a diferença de gaussianas (DDG). Feito isso, uma localização de ponto-chave em que os candidatos a pontos-chave são identificados e refinados, eliminando os pontos de baixo contraste. Em terceiro lugar, uma orientação de atribuição de ponto-chave baseada no gradiente da imagem e por último um gerador de descrição é usado para calcular o descritor de imagem local para cada ponto-chave com base na magnitude do gradiente de imagem e orientação.

Figura 26 - Estágios da seleção de pontos-chave. (a) A imagem original de 233×189 pixels. (b) Os 832 pontos-chave iniciais baseados no máximo e no mínimo da função de diferença de Gauss. Os pontos-chave são exibidos como vetores, indicando escala, orientação e localização. (c) Após aplicar um limite no contraste mínimo, restam 729 pontos-chave. (d) Os 536 pontos chave finais que permanecem após o limiar na relação de curvaturas principais.



Fonte: (LOWE, 2004).

Embora o SIFT tenha provado ser muito eficiente em aplicações de reconhecimento de objetos, ele requer uma grande complexidade computacional, o que o é uma grande desvantagem, especialmente para aplicações em tempo real (KARAMI et al., 2017). Existem diversas variantes e extensões do SIFT que melhoraram sua complexidade computacional (GÜZEL, 2015).

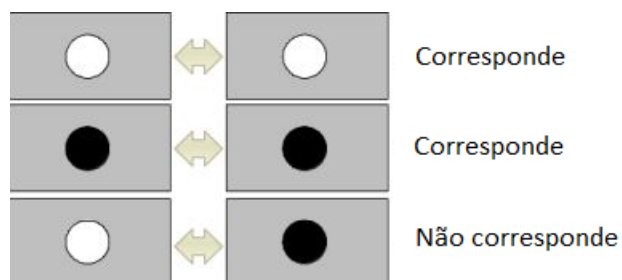
Figura 27- Imagem com pontos-chave e sua orientação usando SIFT



Fonte: OpenCV. Disponível em: <http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html>. Acesso em 13 de set. 2017.

Speeded Up Robust Features (SURF): O SURF é uma versão mais rápida que o SIFT. Ele também é detector e descritor de características, e é baseado em Matriz Hezziana (o SURF também é conhecido como *Fast-Hessian feature detector*). Como descritor ele se encarrega de descrever uma distribuição de *Wavelet* de *Haar*. O descritor SURF é baseado em propriedades similares de informações locais e distribuição gradiente, assim como o SIFT. São usadas 64 dimensões, reduzindo assim o tempo de processamento. O SURF aproxima a DDG com filtros de caixa, quadrados são usados para aproximação já que a convolução com quadrados é muito mais rápida se a imagem integral for usada. Uma região em torno do ponto chave é selecionada e dividida em sub-regiões e, em seguida, para cada sub-região as respostas *wavelet* são tomadas e representadas para obter descritor de recurso SURF. A transformada de Laplace que é já calculada na detecção é usada para pontos de interesse subjacente. No caso da classificação dos recursos é feito uma comparação somente se eles tiverem o mesmo tipo de contraste (baseado na transformada) que permite uma correspondência mais rápida.

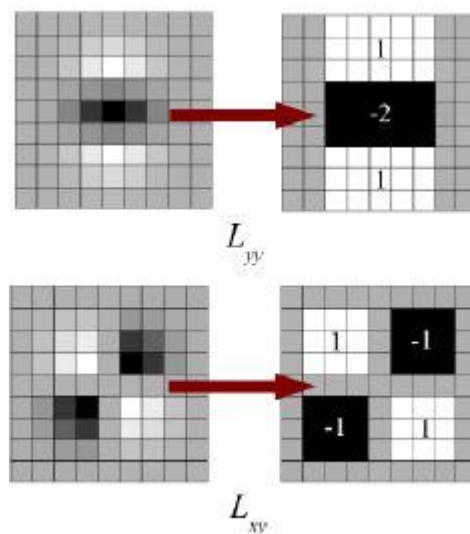
Figura 28- A transformada de Laplace distingue círculos brilhantes em fundos escuros da situação inversa



Fonte: Adaptado de: Introduction to SURF. Disponível em: <https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html>. Acesso em 12 de fev. 2018.

Uma grande vantagem dessa aproximação é que a convolução com filtro de caixa pode ser facilmente calculada com a ajuda de imagens integrais. E isso pode ser feito em paralelo para diferentes escalas.

Figura 29- Convolução com filtro de caixa



Fonte: Adaptado de: Introduction to SURF. Disponível em: <https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html>. Acesso em 12 de fev. 2018.

Binary Robust Independent Elementary Features (BRIEF): Primeiro descritor binário publicado. O BRIEF gera *strings* binárias da comparação do valor de intensidade de cada pixel sobre um *patch* de imagem suavizada usando um núcleo gaussiano. Os patches são obtidos com o *Fast-Hessian feature detector*. O vetor gerado por padrão pelo BRIEF é de tamanho de 256, mas também pode gerar vetores de tamanho 128 e 512.

Oriented FAST and Rotated BRIEF (ORB): Alternativa eficiente para SIFT e SURF, o ORB combina e estende os conceitos do FAST e BRIEF, reduzindo o ruído e obtendo invariância rotacional. O descritor ORB usa processo de aprendizagem para determinar o arranjo espacial de testes binários. O algoritmo de aprendizagem procura por um arranjo de 256 testes não relacionados, que produzem uma *string* de 256 bits. O algoritmo ORB utiliza inicialmente para o FAST para determinar os pontos-chave. Então uma medida de é aplicada para encontrar os melhores N pontos. O FAST não calcula a orientação e é variante de rotação. Calcula a intensidade centroide ponderada do patch com canto localizado no centro. A direção do vetor desse ponto de canto para o centroide dá a orientação. No ORB, uma matriz de rotação é calculada usando a orientação do patch e, em seguida, os descritores BIEF são direcionados de acordo com a orientação.

Binary Robust Invariant Scalable Keypoints (BRISK): Usa o padrão de amostragem simétrica para testes de intensidade. O detector BRISK é baseado no detector AGAST (*Adaptative and Generic Accelerated Segment Test*), que é uma extensão de uma versão mais rápida do detector FAST. Para descrever as características, pares de pixels em torno de pontos de interesse são separados em dois subconjuntos: curta e longa distância. O vetor resultante da operação possui 64 dimensões.

Figura 30- Imagem com pontos-chave calculados pelo algoritmo BRISK.



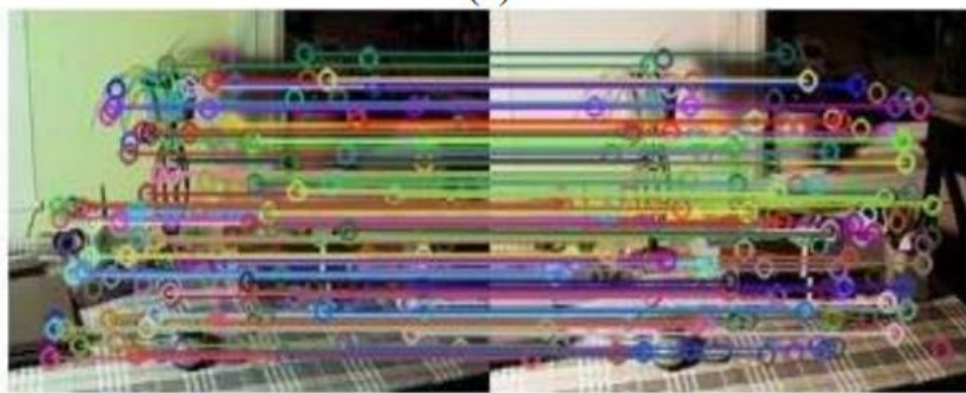
Fonte: Adaptado de Math Works. Disponível em:
<<https://www.mathworks.com/help/vision/feature-detection-and-extraction.html>>.
Acesso em 15 de set. 2017.

Fast Retina Keypoint (FREAK): Inspirado no sistema visual humano, FREAK usa um padrão de retina para testes de intensidade. Semelhante ao BRISK, ele também aplica o detector AGAST. Um algoritmo de aprendizagem semelhante ao ORB é aplicado para encontrar os 512 pares mais relevantes e construir a cadeia de bits.

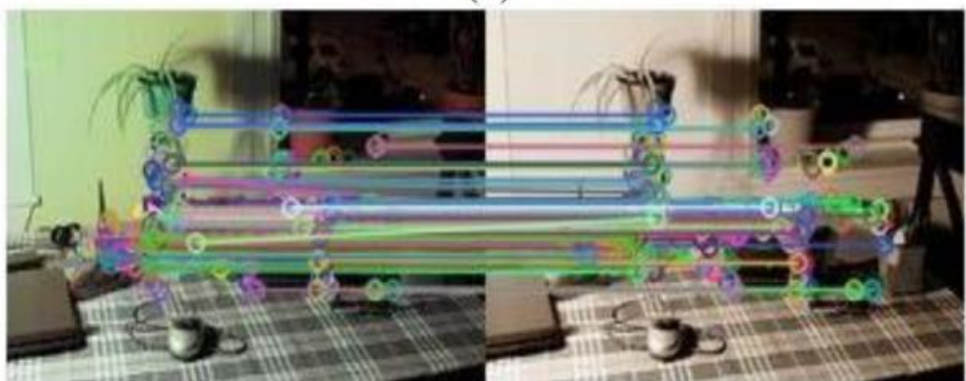
Figura 31- Classificação de imagens de intensidade variável usando (a) SIFT, (b) SURF e (c) ORB



(a)



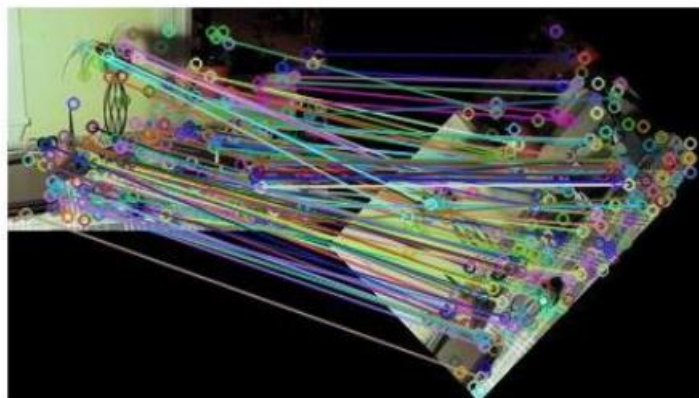
(b)



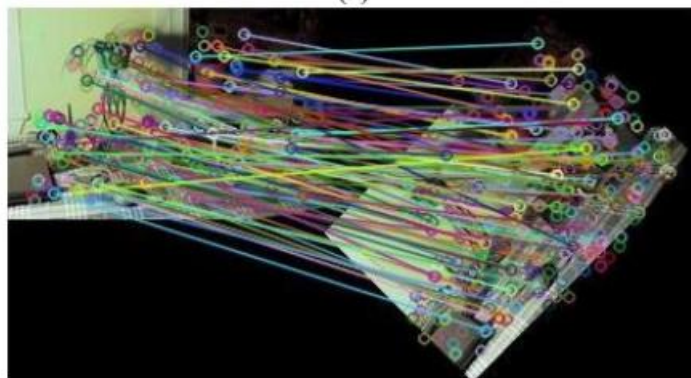
(c)

Fonte: (KARAMI et al., 2017).

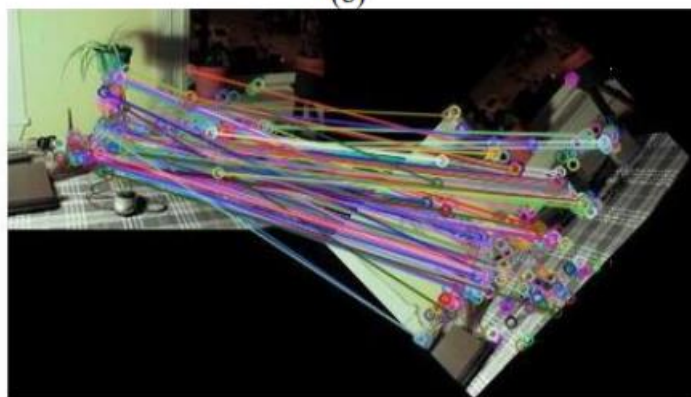
Figura 32- Correspondência da imagem original com a mesma imagem rotacionada usando (a) SIFT, (b) SURF e (c) ORB



(a)



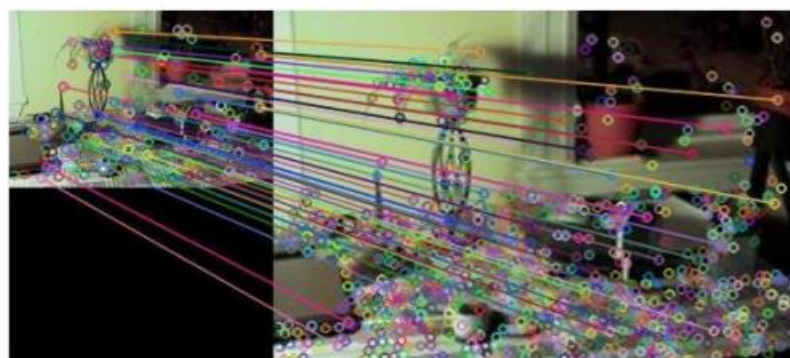
(b)



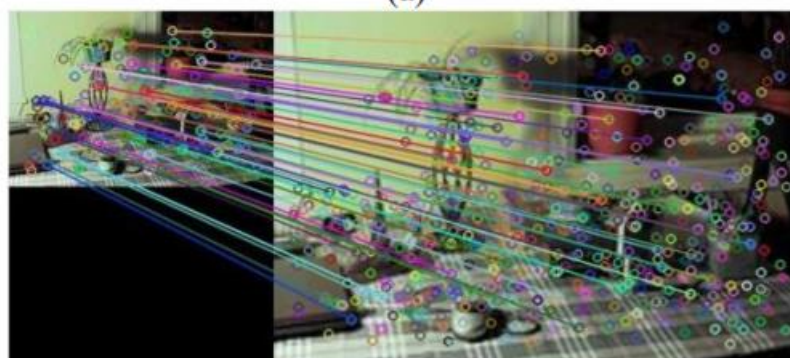
(c)

Fonte: (KARAMI et al., 2017).

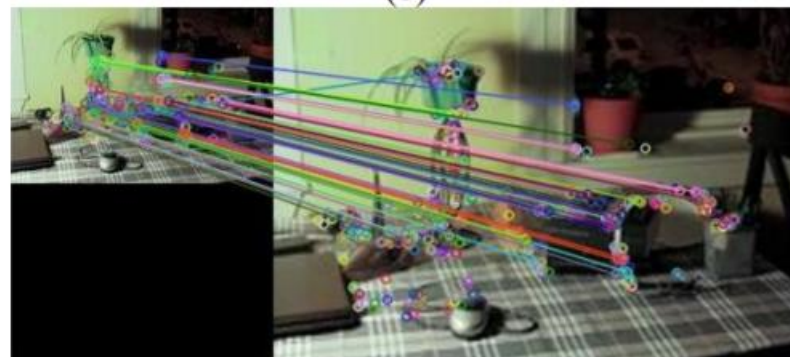
Figura 33- Correspondência da imagem original com a mesma imagem com escala diferente usando (a) SIFT, (b) SURF e (c) ORB



(a)



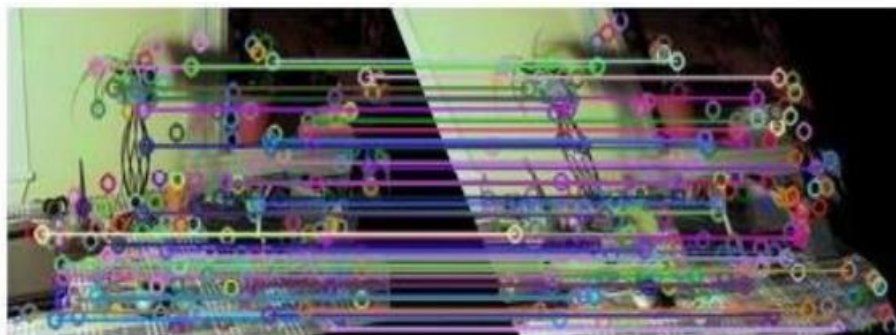
(b)



(c)

Fonte: (KARAMI et al., 2017).

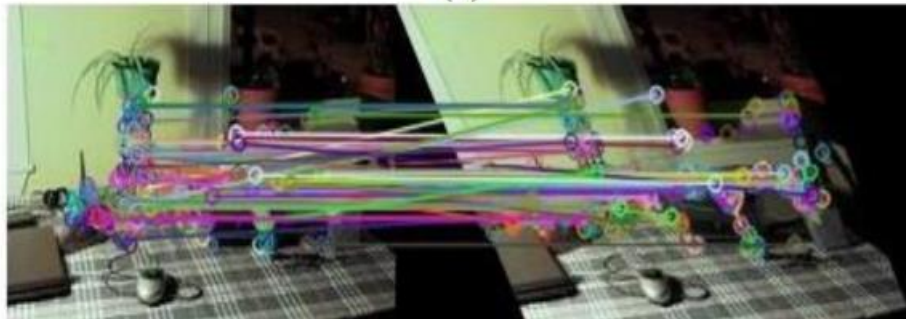
Figura 34- Correspondência da imagem original com a mesma imagem distorcida usando (a) SIFT, (b) SURF e (c) ORB



(a)



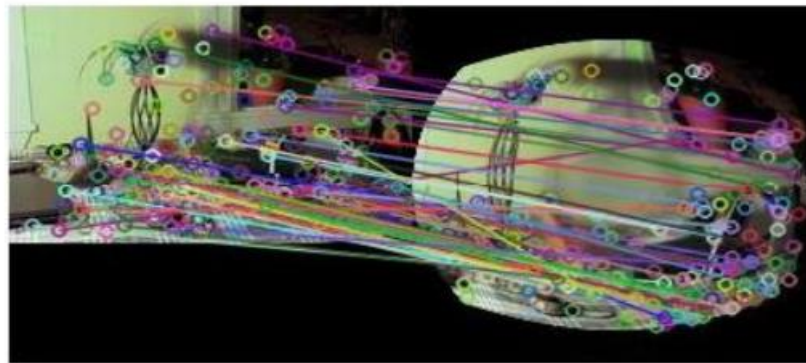
(b)



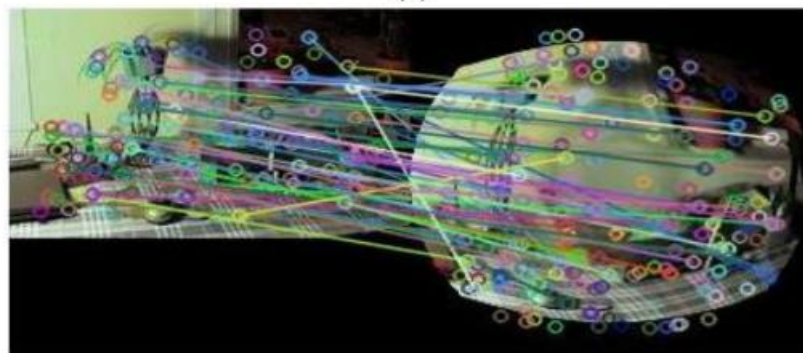
(c)

Fonte: (KARAMI et al., 2017).

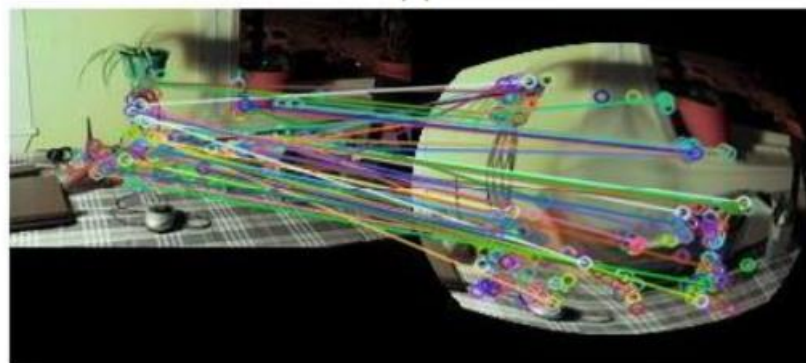
Figura 35- Correspondência da imagem original com a mesma imagem com efeito “olho de peixe” usando (a) SIFT, (b) SURF e (c) ORB



(a)



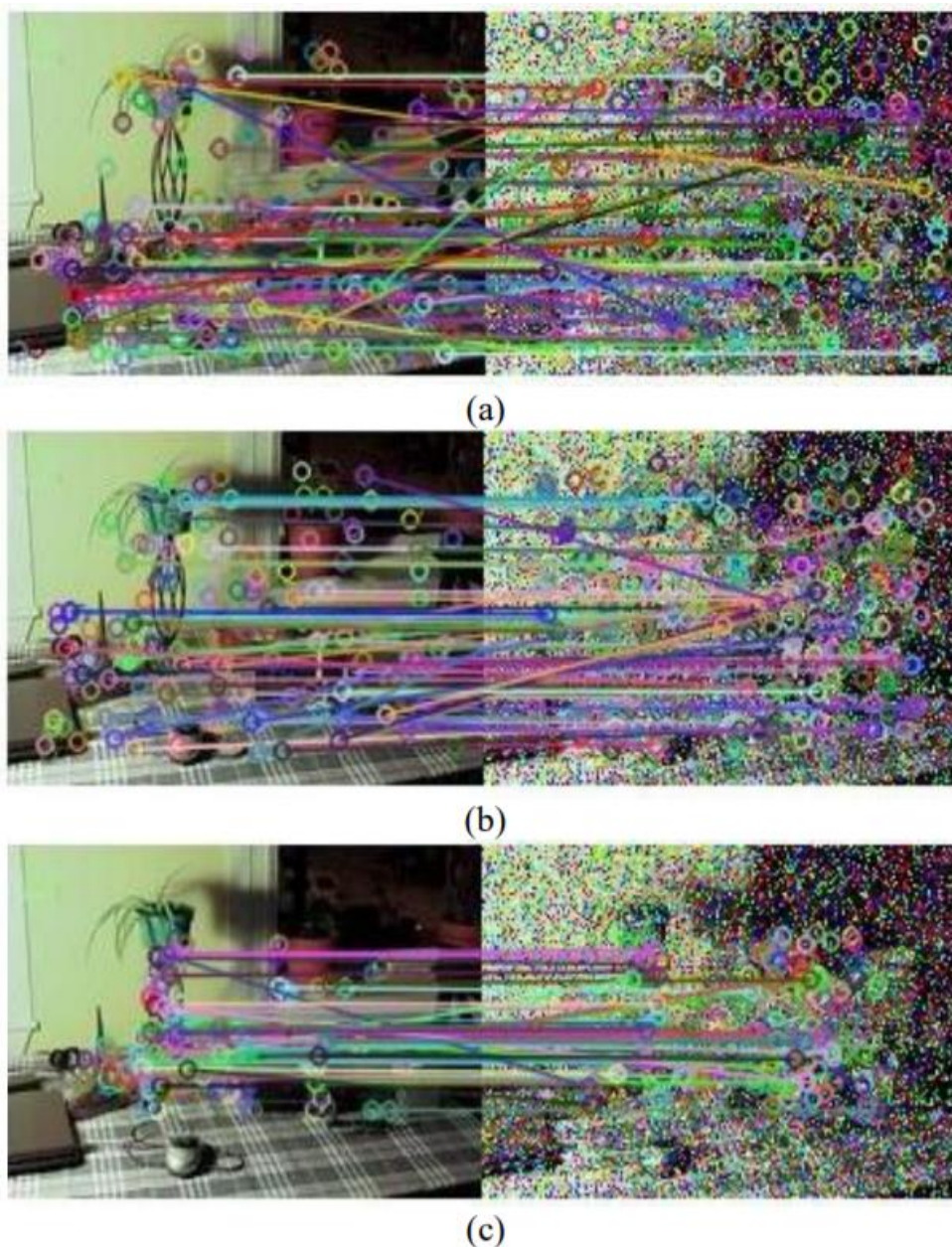
(b)



(c)

Fonte: (KARAMI et al., 2017).

Figura 36 - Correspondência da imagem original com a mesma imagem com efeito “sal e pimenta” usando (a) SIFT, (b) SURF e (c) ORB



Fonte: (KARAMI et al., 2017).

A etapa de classificação consiste na comparação das informações obtidas na etapa de extração de características com o modelo previamente criado. O modelo é gerado por algoritmo de aprendizado de máquina. A aprendizagem de máquina explora o estudo e a construção de algoritmos que podem aprender e fazer previsões sobre dados (KOHAVI et al., 1998).

O aprendizado pode ser dividido em dois tipos principais: supervisionado e não-supervisionado. No aprendizado supervisionado apresenta o conhecimento por conjuntos de exemplos na forma: entrada, saída desejada, ou seja, a máquina é persuadida a aprender. O algoritmo de aprendizado de máquina extrai a representação do conhecimento a partir desses exemplos. Dessa forma, o sistema é capaz de informar a saídas corretas para entradas não apresentadas anteriormente. No aprendizado não-supervisionado não existem exemplos rotulados, dessa forma, o algoritmo é livre para realizar as distinções. O algoritmo de aprendizagem de máquina aprende a representar (ou agrupar) as entradas submetidas segundo uma medida de qualidade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem no entendimento dos dados (DE SOUTO et al.,2003).

Figura 37- Diferentes técnicas de processamento de imagens



Fonte: O autor.

Nota: 1: Na classificação de imagem, categorizamos um objeto alvo. Na classificação e localização, encontramos o objeto alvo e o categorizamos. Na detecção de objetos, buscamos encontrar e classificar os diferentes objetos em uma imagem. Por fim, na segmentação de instâncias, fazemos o particionamento de uma imagem em múltiplos segmentos com o objetivo de simplificar a representação de uma imagem em algo que é mais significativo e mais fácil de ser analisado.

2.3.7 CLASSIFICAÇÃO

A última etapa do SVA consiste na classificação da imagem, ou seja, aplicar um rótulo a um objeto baseado em seus aspectos, encontrados por seus descritores. Podemos destacar o uso de algoritmos de aprendizagem de máquina, dos quais são responsáveis

por analisam os dados gerados pelos descritores, buscando encontrar neles um padrão de informações que possam servir para distinguir os diferentes objetos a serem classificados.

O reconhecimento de padrões em imagens é uma área do campo de processamento de imagens que visa a classificação de objetos dentro de um número de categorias, e é amplamente utilizado, podemos citar alguns exemplos como:

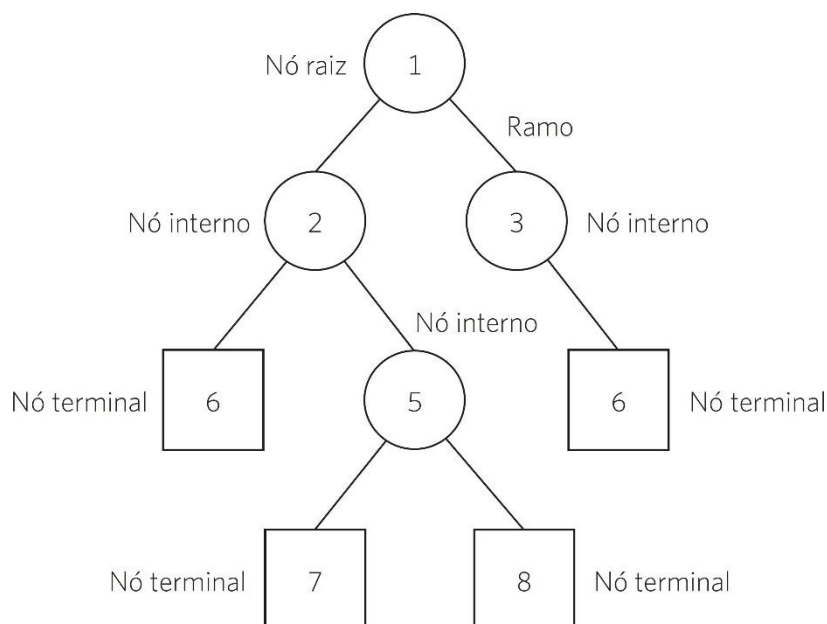
- Reconhecimento de caracteres
- Reconhecimento de impressões digitais
- Reconhecimento de assinaturas
- Reconhecimento de faces

2.3.7.1 ÁRVORE DE DECISÃO

Uma árvore de decisão é uma estrutura de nodos conectados por ramos, onde os nodos são atributos classificatórios e os ramos correspondentes descendentes entendidos como possíveis valores. Cada nodo folha recebe um rótulo de classe, com isso, os caminhos completos da árvore formam as regras de classificação (TAN; STEINBACH; KUMAR, 2009, RUIZ; CATEN; DALMOLIN, 2014).

Estes modelos utilizam a estratégia de dividir para conquistar: um problema complexo é dividido em problemas menos complexos e recursivamente esta técnica é aplicada a cada subproblema. As árvores de decisão estão entre os mais populares algoritmos de inferência e tem sido aplicado em várias áreas como, por exemplo, diagnóstico médico e risco de crédito, e deles pode-se extrair regras do tipo “se-então” que são facilmente compreendidas.

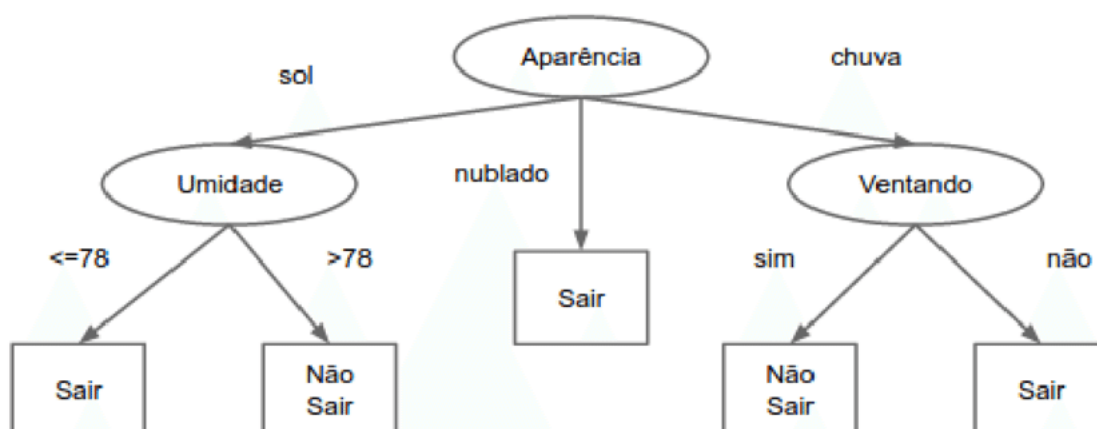
Figura 38 - Estrutura geral de uma árvore de decisão



Fonte: (MEDEIROS et al., 2014)

Nota: 1: nó raiz ou principal (com atributo preditivo). Nós internos ou filhos (com variáveis decisórias de níveis intermediários): 2, 3 e 5. Folhas ou nó terminal (com o valor da variável de decisão): 4, 6, 7 e 8. Ramos são as conexões entre os nós que contêm os valores dos atributos de cada variável decisória.

Figura 39 - Representação de árvore de decisão



Fonte: (RIBEIRO, 2015).

Nota: 1: Árvore de decisão para "Sair" ou "Não sair". O nó raiz é a aparência do tempo, em caso de sol, o próximo atributo a ser analisado é a umidade, e em caso dela ser ≤ 78 , o resultado da tomada de decisão

será “Sair”, caso contrário o resultado será “Não sair”. Já quando a aparência for nublada, a tomada de decisão indica para “Sair”. Ao constatarmos que a aparência é de chuva, e seu atributo filho for positivo quanto a ventando, a tomada de decisão indicará para “Não sair”, e caso o atributo ventando seja negativo, a tomada de decisão indicará para “Sair”.

A construção de uma árvore de decisão baseia-se em três características, são elas: reduzir a entropia (a aleatoriedade da variável objetivo), ser consistente com o conjunto de dados e possuir o menor número de nós.

O aprendizado indutivo de árvores de decisão é geralmente dividido em aprendizado supervisionado e não-supervisionado, embora o aprendizado semi-supervisionado também tem sido considerado ao longo dos últimos anos (CHAPELLE et al., 2006).

A árvore de decisão é montada a partir de atributos para a classificação, onde quanto maior a utilidade do atributo, maior é o ganho de informação. Nos casos em que a árvore é usada para classificação, os critérios de partição mais conhecidos são baseados na entropia e índice Gini.

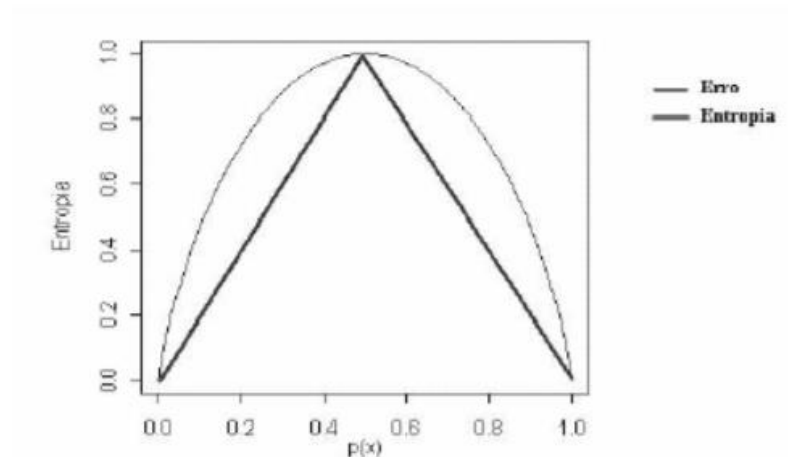
Entropia é o cálculo do ganho de informação baseado em uma medida utilizada na teoria da informação. A entropia representa o quão puros ou impuros estão os dados. A entropia é máxima (igual a 1) quando o conjunto de dados é heterogêneo. Dado um conjunto de entrada (S) que pode ter c classes distintas, a entropia de S será dada por:

$$\text{Entropia (S)} = \sum_{i=1}^c -p_i \log_2 p_i$$

Onde:

- P_i é a quantidade de dados em S pertencentes à classe i.
- Entropia máxima ($\log_2 c$) se $p_i = p_j \forall i \neq j$
- Entropia(S) = 0 se $\exists i$ tal que $p_i = 1$.
- Por hipótese, $0 \cdot \log_2 0 = 0$

Figura 40 - Representação gráfica $p(x)$ versus entropia



Fonte: O autor.

O índice Gini foi desenvolvido por Conrado Gini em 1912, onde mede-se o grau de heterogeneidade dos dados. Logo, pode ser utilizado para medir a impureza de um nó.

O índice Gini é representado por:

$$\text{Índice Gini} = 1 - \sum_{i=1}^c p_i^2$$

Onde:

- P_i é a frequência relativa de cada classe em cada nó.
- c é o número de classes.

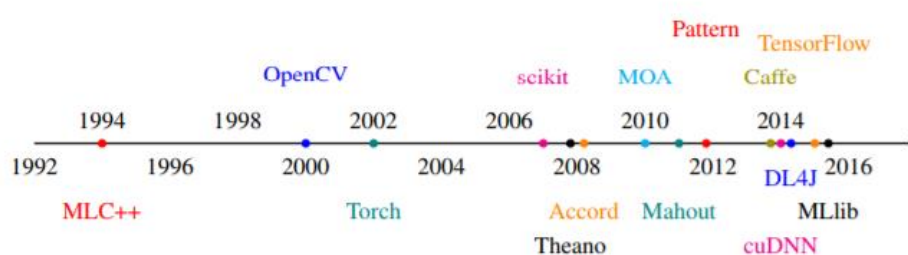
Dessa forma, concluímos que quando este índice é igual a zero, o nó é homogêneo. Já quando ele se aproxima do valor um, o nó é impuro (aumenta o número de classes uniformemente distribuídas neste nó). Ao utilizarmos o critério de Gini nas árvores de classificação binárias, tende-se a isolar num ramo os registros que representam a classe mais frequente. Quando se utiliza a entropia, balanceia-se o número de registros em cada ramo.

2.4 MACHINE LEARNING

Implementação de algoritmos genéricos que podem extrair informação de conjunto de dados sem a necessidade de código específico ao problema. Os algoritmos genéricos constroem sua própria lógica baseados nos dados com que são alimentados. Por exemplo, algoritmos genéricos cuja função seja classificar dados, podem ser usados para diferentes grupos, como um algoritmo de classificação de manuscritos que também pode ser treinado para classificar imagens sem alteração de seu código, dessa forma os algoritmos genéricos podem ser reutilizados em diferentes problemas de classificação.

A aprendizagem de máquina é dividida em categorias distintas, sendo elas as mais importantes: aprendizagem supervisionada e aprendizagem não supervisionada, onde na aprendizagem supervisionada informamos ao algoritmo qual é a resposta de determinado problema, por exemplo, treinamos o algoritmo com imagens de carros e informamos a ele que essas imagens são de carros, caberá então ao algoritmo realizar uma engenharia reversa para descobrir qual é a lógica por trás dessa informação. Já na aprendizagem não supervisionada, não informamos ao algoritmo qual é a resposta para os dados de entrada, fazendo com que ele fique responsável de encontrar um padrão ou agrupamento dos dados de saída.

Figura 42 - Linha do tempo mostrando os lançamentos de bibliotecas de *machine learning*



Fonte: (GOLDSBOROUGH, 2016)

2.5 DEEP LEARNING

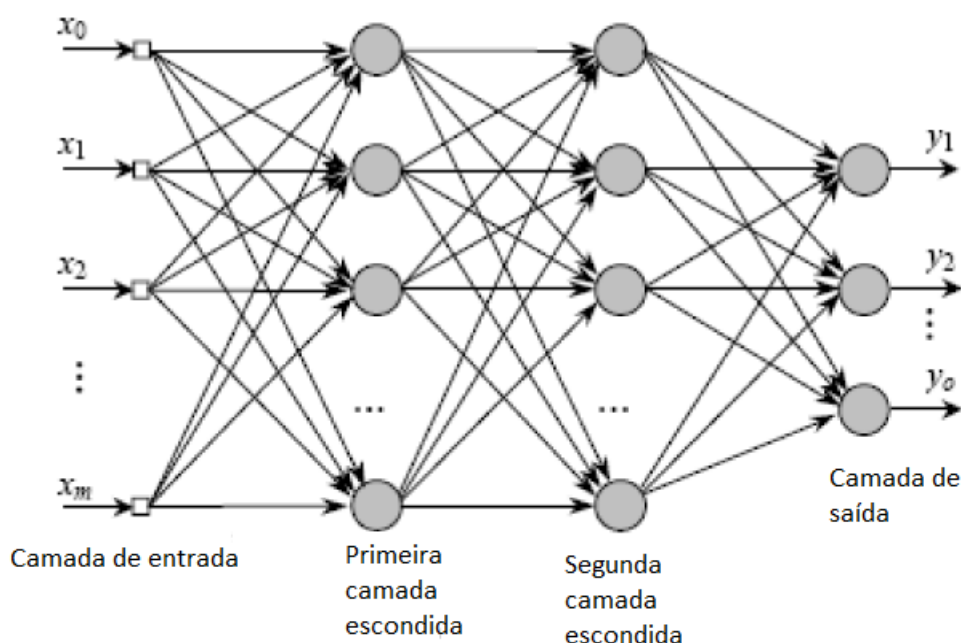
O termo “*Deep Learning*” do inglês “Aprendizagem Profunda” permite que modelos computacionais compostos de múltiplas camadas de processamento aprendam representações de dados com múltiplos níveis de abstração (LECUN et al., 2015). O uso de métodos de *Deep Learning* melhoraram significativamente o estado da arte em diversas áreas distintas, como reconhecimento de fala, reconhecimento de objetos visuais,

detecção de objetos e muitos outros domínios, como descoberta de drogas e estudo do genoma. O aprendizado profundo é capaz de encontrar estruturas complexas em grandes conjuntos de dados usando o algoritmo de retropropagação para indicar como uma máquina deve alterar seus parâmetros internos que são usados para calcular a representação em cada camada da representação na camada anterior.

Novos algoritmos foram capazes de ultrapassar o nível humano de precisão no reconhecimento de objetos, fazendo da aprendizagem profunda dos tópicos mais abordados em ciência da computação nos últimos anos (LECUN et al., 2015). Podemos destacar grandes empresas do setor de TI que estão investindo fortemente nesse tipo de tecnologia através do desenvolvimento e busca de popularização de *frameworks* de *deep learning*, como a Microsoft por meio do seu *Cognitive Toolkit*, anteriormente conhecido pela sigla CNTK, a IBM com o Watson, e o Google disponibilizando o TensorFlow. Em suma, essas empresas fornecem plataformas estáveis para pesquisa e aplicações de *deep learning* a fim de facilitar o uso e reprodutibilidade de seus métodos.

Segundo Rampasek e Goldenberg (RAMPASEK; GOLDENBERG, 2016), um modelo de aprendizagem profunda, tipicamente um rede neural multicamada, é composta de várias camadas computacionais que processam dados de forma hierárquica. Cada camada recebe uma entrada e produz um saída, muitas vezes computada como uma função não-linear de uma combinação linear ponderada dos valores de entrada. A saída de uma camada torna-se a entrada para a próxima camada de processamento, criando uma arquitetura profunda, conforme figura abaixo:

Figura 41- Rede neural artificial multicamadas



Fonte: (DE OLIVEIRA et. al, 2010)

Ao passar pelas camadas, os dados são representados de uma maneira cada vez mais abstrata. Particularmente populares são as "camadas convolucionais" que aplicam uma função local, também chamada de filtro, para todos os subconjuntos da camada de entrada, como as partes de uma imagem. Podemos tomar como exemplo a tarefa de reconhecimento de objetos, onde as primeiras camadas representam formatos básicos como linhas, curvas e outras primitivas visuais. Filtros criados à partir dessas primitivas são aprendidos pelo modelo. As camadas superiores ficam a cargo de formas mais complexas, por exemplo, um rosto humano. Esses modelos profundos podem aprender uma abstração de dados altamente complexa que captura a intrincada estrutura de conjuntos de dados muito grandes. (RAMPASEK; GOLDENBERG, 2016). A unidade básica de computação das redes neurais é chamada neurônio. Um neurônio recebe uma entrada x onde efetua um cálculo sobre ela para produzir um valor. Este valor é passado para uma função não linear chamada função de ativação (f) para produzir a saída final ou também chamada de ativação de um neurônio. Existem muitos tipos de funções de ativação. Uma das funções de ativação populares é Sigmóide:

$$y = \frac{1}{1 + e^{-z}}$$

O conjunto de neurônios agrupados em linha, é chamado de camada, que é a base da construção das redes neurais. A primeira camada da rede pela qual os dados de entrada são passados para a rede é chamada de camada de entrada. Da mesma forma, a última camada é chamada de camada de saída. As camadas entre as camadas de entrada e saída são chamadas de camadas ocultas ou camadas escondidas. As redes que têm muitas camadas ocultas tendem a ser mais precisas e são chamadas de rede profunda e, portanto, algoritmos de aprendizado de máquina que usam essas redes profundas são chamadas de aprendizado profundo.

Ainda de acordo com os autores Rampasek e Goldenberg, a biologia computacional, no campo da predição de estrutura de proteínas foi dos primeiros a adotar a aprendizagem profunda, aplicando desde os anos 80. Por exemplo, um modelo de aprendizagem profunda de 50 camadas melhorou as previsões no conjunto de dados CASP8 em 10% (DI LENA et al., 2012 apud RAMPASEK; GOLDENBERG, 2016).

Após o desafio MERCK 2012 de atividade molecular, a *deep learning* recebeu uma maior atenção no campo de biologia computacional, quando um modelo profundo foi o grande vencedor da competição (DAHL et al., 2014 apud RAMPASEK; GOLDENBERG, 2016). Vale ressaltar que a equipe vencedora tinha conhecimento apenas na área de aprendizagem de máquina e não em biologia molecular. Podemos citar também que a *deep learning* obteve resultados satisfatórios em imagiologia médica, na segmentação de imagens do cérebro para ajudar a diagnosticar a doença de Alzheimer.

2.6 REDE NEURAL CONVOLUCIONAL

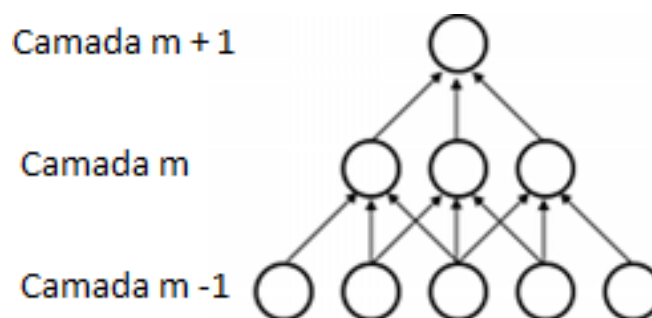
Rede Neural Convolutacional (do inglês *Convolutional Neural Network* – CNN, ou conhecida também por ConvNet) é uma classe de rede neural artificial que vem sendo aplicada com sucesso no processamento e análise de imagens digitais. Essa classe é uma variante da rede Perceptrons de Múltiplas Camadas, e foi inspirada no processo biológico de processamentos de dados visuais. As células responsáveis pelo processamento de dados visuais são sensíveis a pequenas sub-regiões do campo visual, também chamado de campo receptivo. Essas células atuam como filtros locais sobre o campo de visão e são adequadas para explorar a correlação espacial local presente em imagens naturais. Os campos receptivos de diferentes neurônios se sobrepõem parcialmente de forma a cobrir todo o campo de visão. A CNN tira proveito das cores. Utilizar imagens RGB em vez de escala de cinza no treinamento, pode ocasionar em uma melhora significativa no

resultado. Nos humanos, a cor ajuda o sistema visual a analisar imagens complexas com mais eficiência, melhorando o reconhecimento de objetos. Alguns experimentos psicológicos mostraram como a cor dá uma forte contribuição à memorização e reconhecimento (WICHMANN; SHARPE, 2002).

As CNNs são capazes de aplicar filtros em dados visuais, mantendo a relação de vizinhança entre os pixels da imagem ao longo do processamento da rede. Este tipo de rede vem sendo amplamente utilizada, principalmente nas aplicações de classificação, detecção e reconhecimento em imagens e vídeos.

As CNNs analisam a correlação espacial local impondo um padrão de conectividade local entre os neurônios das camadas próximas. Em outras palavras, as entradas de unidades ocultas na camada m são de um subconjunto de unidades na camada $m-1$, unidades que possuem campos receptivos espacialmente contíguos:

Figura 42- Representação de conexão de camadas de rede neural



Fonte: O autor.

Tomemos como exemplo a figura acima, onde a camada $m-1$ representa a retina (entrada). As unidades na camada m têm campos receptivos de largura 3 e, portanto, são conectados a apenas 3 neurônios na camada superior. As unidades na camada $m + 1$ têm uma conectividade semelhante com a camada m . Dizemos que seu campo receptivo em relação à camada m também é 3, mas seu campo receptivo em relação ao input é maior (5). Cada unidade não responde a variações fora de seu campo receptivo em relação à retina. A arquitetura garante, assim, que os "filtros" aprendidos produzam a resposta mais forte a um padrão de entrada espacialmente local.

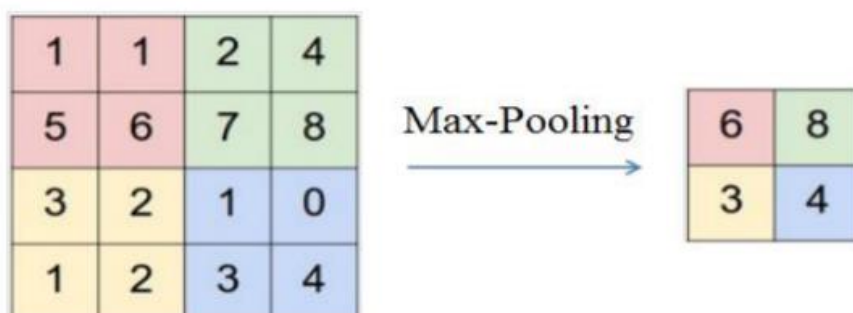
O empilhamento de muitas dessas camadas resulta na criação de "filtros" que respondem a uma região maior do espaço de pixels. Por exemplo, a unidade na camada

oculta $m + 1$ pode codificar um recurso não linear de largura 5 (em termos de espaço de pixels).

Uma rede neural convolucional consiste em múltiplas camadas com funções diferentes. Inicialmente é comum aplicar sobre o dado de entrada as camadas que dão nome à rede neural, chamadas de convolução. Uma camada de convolução é composta por diversos neurônios, cada um responsável por aplicar um filtro em uma região da imagem. Cada neurônio é conectado a um conjunto de pixels da camada anterior e cada uma dessas conexões se atribui um peso. A combinação das entradas de um neurônio, utilizando os pesos respectivos de cada uma de suas conexões, produz uma saída passada para a camada seguinte. O filtro de convolução é representado por uma matriz que contém os pesos atribuídos as conexões de um neurônio.

Outras funções que as redes convolucionais podem incluir são camadas de agrupamento locais ou globais (também conhecidas como *pooling layers*), que combinam as saídas de agrupamentos de neurônios em uma camada em um único neurônio na próxima camada. Por exemplo, o pooling máximo usa o valor máximo de cada cluster de neurônios na camada anterior. Outro exemplo é o *pooling* médio, que usa o valor médio de cada um dos grupos de neurônios na camada anterior. Em resumo essa camada reduz a dimensionalidade de um mapa de característica fornecido como entrada e produz outro mapa de característica, uma espécie de resumo do primeiro (HIJAZI, 2015).

Figura 43- Max-Pooling com um filtro de tamanho 2x2 e tamanho do passo (Stride)



Fonte: (HIJAZI, 2015).

A função das camadas de *dropout* (NITISH et. Al, 2014) é desativar um conjunto aleatório de ativações nessa camada, definindo-as como zero. Com isso, a rede deve ser capaz de fornecer a classificação ou a saída correta para um exemplo específico, mesmo

se algumas das ativações forem descartadas. Isso garante que a rede não esteja "ajustada" demais aos dados de treinamento e, assim, ajuda a aliviar o problema de *overfitting*.

As camadas totalmente conectadas ou *Fully Connected Layers* conectam todos os neurônios em uma camada a todos os neurônios em outra camada. É, em princípio, o mesmo que a rede neural perceptron multicamada tradicional.

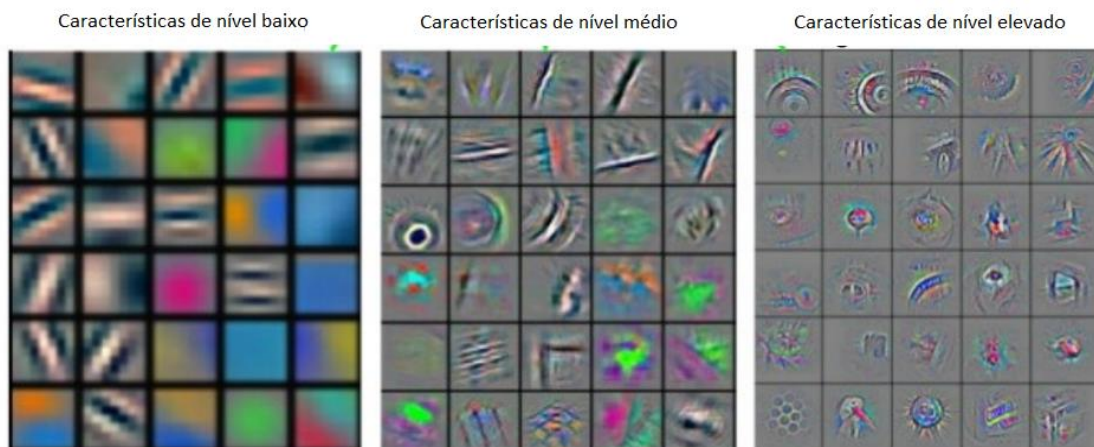
As redes neurais convolucionais fazem a análise de campos receptivos locais (*local receptive fields*). Os neurônios da mesma camada são agrupados em mapas. Um mapa é produzido pelo agrupamento das saídas de neurônios que juntos cobrem uma parte da imagem que tenha sido processado com um filtro em comum. Durante o processo de treinamento, é realizado um compartilhamento dos pesos para que um conjunto de neurônios de um determinado mapa aplique o mesmo filtro em diferentes posições da imagem. Tal compartilhamento reduz significativamente o número de parâmetros a serem aprendidos e o tempo de treinamento da rede, conseqüentemente.

O número de vizinhos que serão processados por cada neurônio da camada é definido pelo tamanho do filtro. A camada convolucional também pode ignorar um número de pixels, este valor, chamado de passo ou *stride* representa quantos pixels serão pulados entre cada janela, dizendo-nos qual será o tamanho da camada seguinte na mesma unidade.

Nas camadas de convolução das CNNs é preciso apenas definir a arquitetura dos filtros: quantidade e tamanho, passo, dos filtros por camada. O processo de aprendizado da rede altera os pesos ao longo do treinamento, até encontrar os melhores valores dos filtros para o conjunto de dados utilizado, tornando assim desnecessária a indicação de qual filtro deve ser utilizado.

Diferentes extratores de características (filtros) emergem em diferentes camadas durante o treinamento da rede, onde em camadas mais baixas, características como linhas, contraste e cor são representadas. Já em camadas médias são representados cantos ou outras conjunções de borda/cor e texturas. E nas camadas mais elevadas, são extraídos filtros mais complexos, específicos da classe.

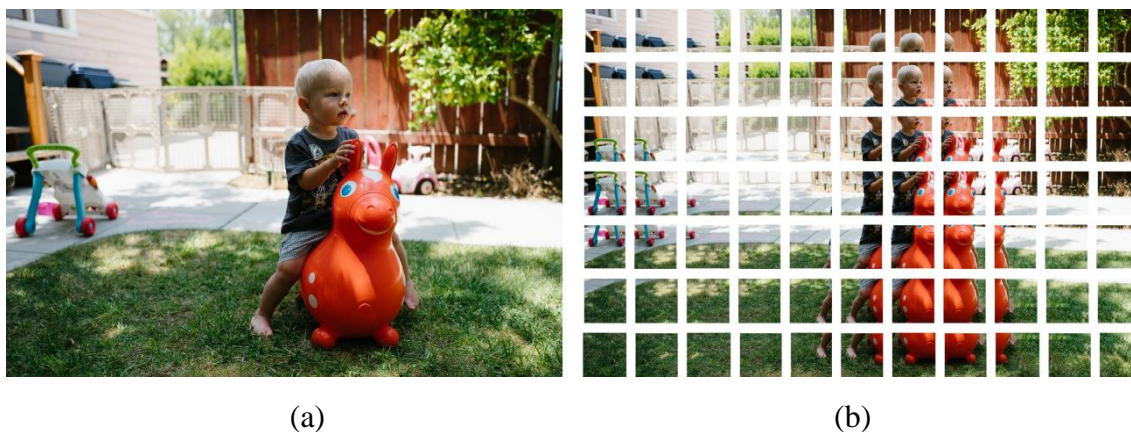
Figura 44 - Extração de características de imagens feita por níveis diferentes de camadas de CNN.



Fonte: (ZEILER; FERGUS, 2014).

O processo de aprendizagem inicia-se ao dividirmos a imagem em recortes sobrepostos e então alimentamos a rede ao invés de passar a imagem inteira como uma grade de números. A ideia é que o objeto é o mesmo não importa onde ele apareça em uma imagem.

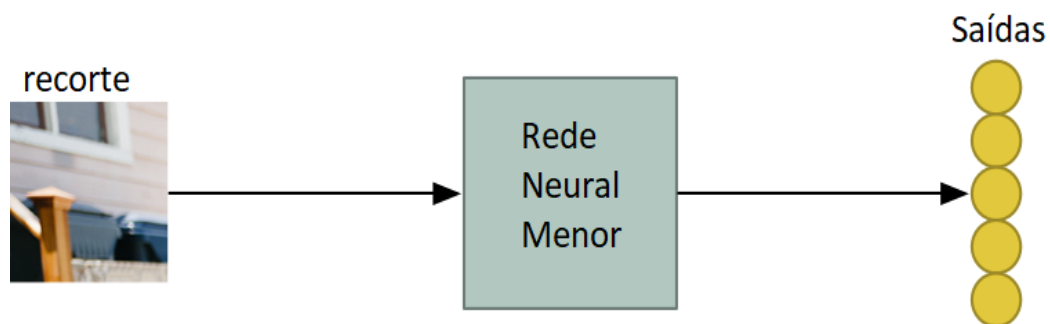
Figura 45- (a) Imagem original. (b) Imagem com recortes sobrepostos.



Fonte: Adaptado de: Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Disponível em <<https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>>. Acesso em: 25 de mar. 2018.

Na figura 42, dividimos a imagem em 77 recortes menores e de mesmo tamanho. O próximo passo é passar cada recorte da imagem para uma rede neural menor, mantendo os mesmos pesos da rede para cada recorte da imagem original, tratando igualmente as informações contidas em cada recorte.

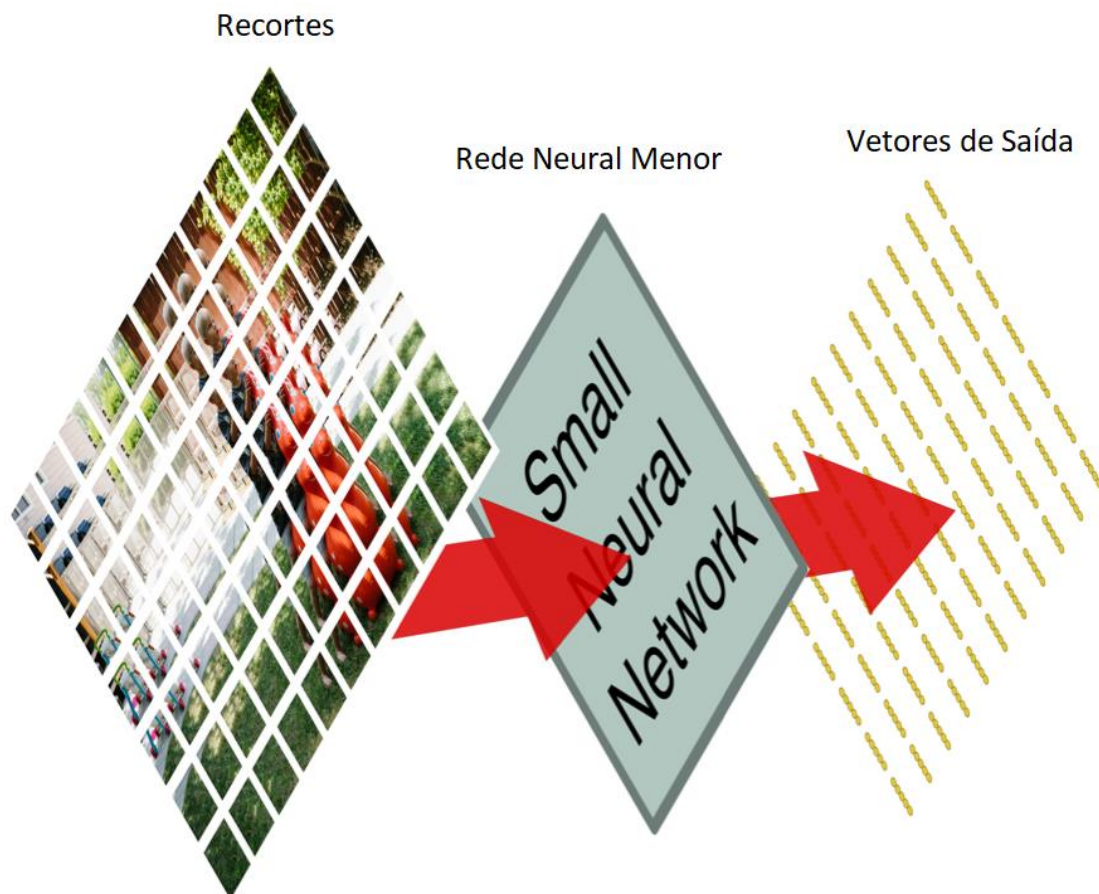
Figura 46- Esquema de processamento de cada recorte



Fonte: Adaptado de: Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Disponível em <<https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>>. Acesso em: 25 de mar. 2018.

Os resultados de saída de cada recorte são salvos em um novo vetor que é inserido em uma matriz organizada de maneira igual à imagem original:

Figura 47 - Representação da organização dos vetores de saída com os recortes da imagem original

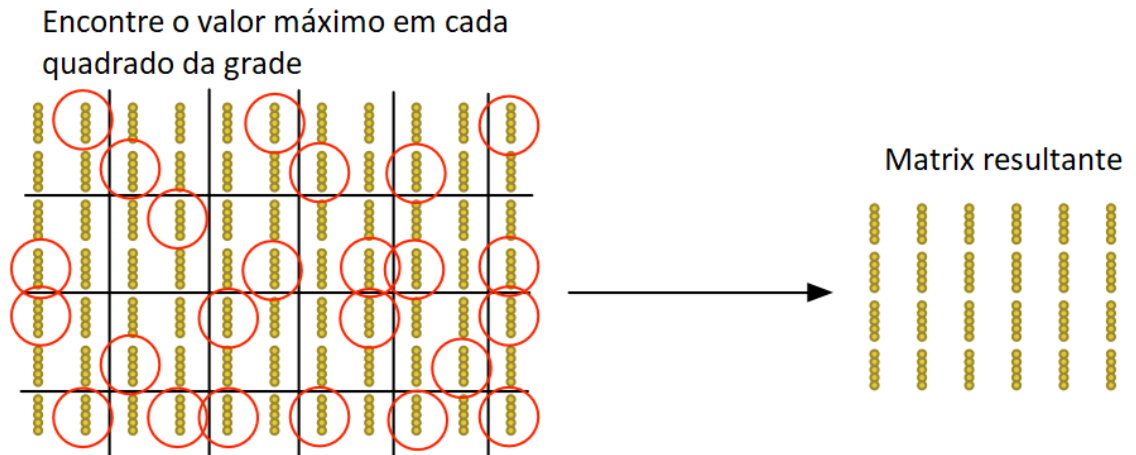


Fonte: Adaptado de: Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Disponível em <<https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>>. Acesso em: 25 de mar. 2018.

Começamos o processo com uma imagem grande e ao final obtemos um vetor menor onde estão registradas quais seções de nossa imagem são as mais interessantes.

Mesmo reduzindo o tamanho das informações quando comparamos o tamanho original da imagem, o vetor ainda é grande, podendo ser reduzido ainda mais. Para reduzirmos o vetor, tiramos uma amostra do mesmo, com um algoritmo chamado *Max Pooling*, esse algoritmo analisa cada quadrado 2x2 do vetor e mantém os números maiores.

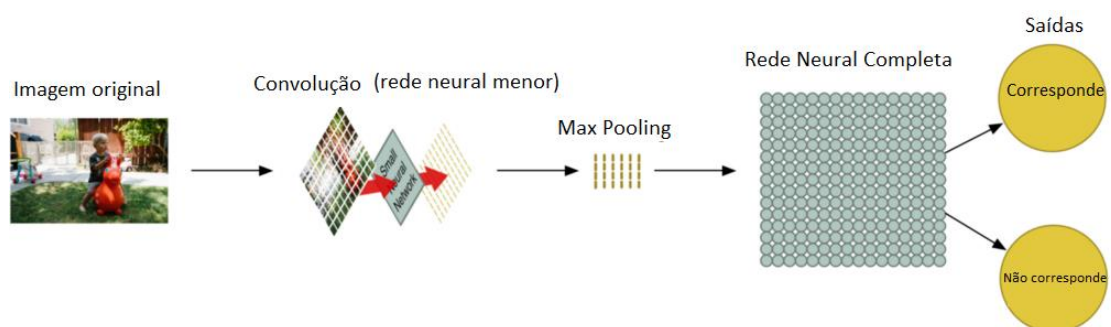
Figura 48- Representação da redução da matriz original após a aplicação do algoritmo de *max pooling*.



Fonte: Adaptado de: Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Disponível em <<https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>>. Acesso em: 25 de mar. 2018.

Mantemos assim apenas o que o algoritmo julgar interessante, e com isso reduzimos o tamanho da matriz. O próximo passo é utilizar a matriz resultante em outra rede neural, para que objetos sejam identificados.

Figura 49- Resumo do processo de classificação por CNN

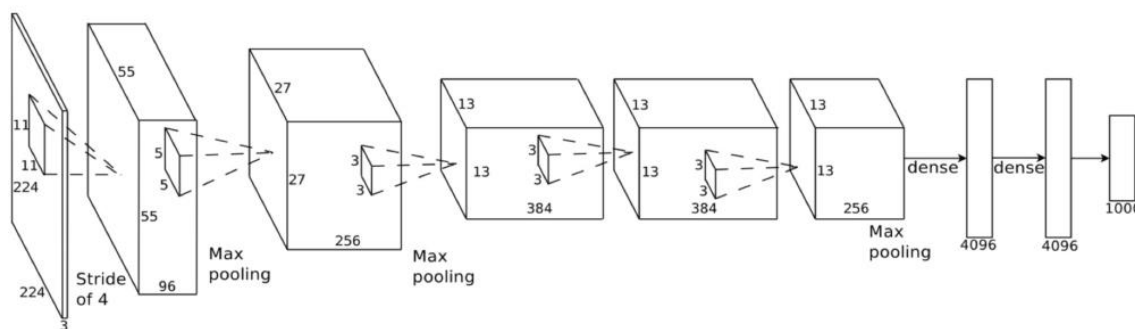


Fonte: Adaptado de: Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Disponível em <<https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>>.

[fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721](https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721)>. Acesso em: 25 de mar. 2018.

Resumindo, a rede neural convolucional aqui exemplificada contém os passos de convolução, *max pooling*, e uma rede completamente conectada ao final. Podemos combinar, empilhar e repetir esses passos conforme a necessidade. Quanto mais passos de convolução usarmos, maior a capacidade da rede de aprender a reconhecer padrões complexos.

Figura 50 - Outro exemplo de rede neural convolucional profunda.



Fonte: Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Disponível em <<https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>>. Acesso em: 25 de mar. 2018.

No exemplo acima, uma imagem de 224x224 pixels é passada para a rede, onde é aplicada convolução e duas *max pooling*, depois disso, a convolução é executada três vezes, seguida de *max pooling* e então duas camadas completamente conectadas são usadas.

2.7 TENSORFLOW

A implementação de um modelo complexo de rede neural não é uma tarefa simples. Essa tarefa foi essencialmente restrita a especialistas na área ou aqueles que acabaram adquirindo conhecimento durante o processo. Graças à estrutura modular das

redes e ferramentas de inferência padrão, vários frameworks surgiram para facilitar e agilizar o design e treinamento de redes neurais, uma dessas ferramentas é o TensorFlow.

O TensorFlow é um sistema de aprendizado de máquina que opera em larga escala e em ambientes heterogêneos (ABADI et al., 2016). Ele fornece várias melhorias, como visualização gráfica e melhor tempo de compilação. Os frameworks de *deep learning* mais amplamente usados hoje são Torch7, Theano e Caffe, que é particularmente adequado para redes neurais convolucionais (RAMPASEK; GOLDENBERG, 2016).

Utilizando do paradigma de programação declarativa, o Tensorflow permite aos pesquisadores se concentrar na definição simbólica do que precisa ser calculado ao invés de como exatamente, e em que ordem particular, diferentemente da programação imperativa. A representação simbólica permite a diferenciação automática, que fornece uma maneira conveniente de otimizar muitas funções. Estas podem ser redes neurais ou outras funções que são comumente usadas para representar matematicamente diferentes problemas orientados por dados. O processamento da representação abstrata do modelo é tratado automaticamente pelo *framework*. Isso faz com que o Tensorflow seja especialmente adequado para o desenvolvimento de novos modelos que usam otimização baseada em gradiente. Outra vantagem do TensorFlow é que vem com uma ferramenta de suporte chamada TensorBoard para visualização do progresso do treinamento do modelo. Dessa forma, podemos acompanhar interativamente a estrutura do gráfico computacional, bem como os parâmetros e modelo desempenho estão mudando ao longo do treinamento de iterações (ABADI et al., 2016).

Tabela 2 - Comparação de características entre Tensorflow, Theano, Torch e Caffe

Biblioteca	Linguagem de programação	Paradigma	Gradiente	Execução distribuída
Tensorflow	Python, C++	Declarativa	Simbólica	Sim
Theano	Python	Declarativa	Simbólica	Não
Torch	LuaJIT	Imperativa	Explicita	Não
Caffe	Protobuf	Imperativa	Explicita	Não

Fonte: O autor.

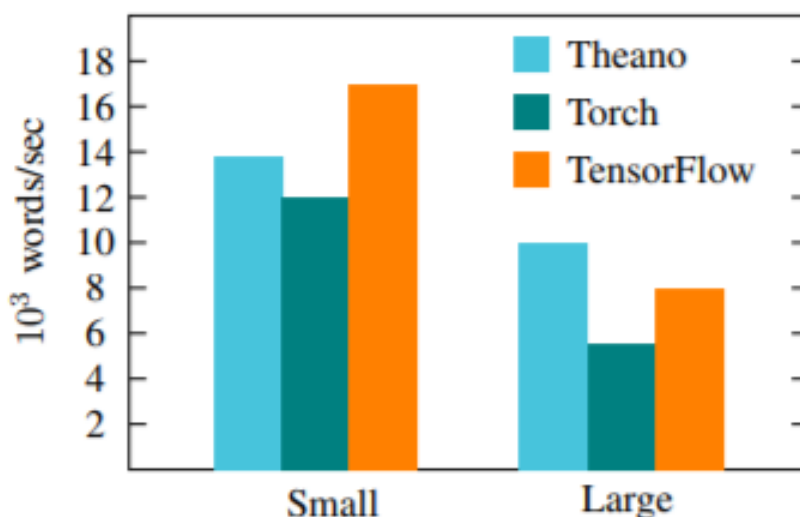
A visualização por meio do Tensorboard fornece uma representação modular de um modelo. Isso facilita a representação global do modelo, a depuração e verificação para melhor compreensão durante o desenvolvimento do modelo. O TensorFlow pode ser executado somente em única máquina. No entanto, é possível fazer a paralelização em vários processadores (CPUs ou GPUs) em uma única máquina.

Wrappers de alto nível do TensorFlow, como Keras ou Pretty Tensor, auxiliam na tarefa de construção da rede neural, fornecendo uma maneira simplificada de construir as redes neurais por padrão de camadas.

O Tensorflow contém algumas características, como:

- *Train accuracy* (Precisão de treinamento): mostra a porcentagem das imagens usadas no treinamento atual que foram rotuladas com a classe correta.
- *Validation accuracy* (Precisão de validação): a precisão de validação é a porcentagem de imagens rotuladas corretamente em um grupo de imagens selecionadas aleatoriamente de um conjunto diferente.
- *Cross entropy* (Entropia cruzada): função de perda que dá uma ideia de quão bem o processo de aprendizado está progredindo. Quanto mais próximo de zero é a entropia cruzada, melhor o resultado.

Figura 51- Comparação entre Tensorflow, Theano e Torch para o conjunto de dados Penn Treebank



Fonte: (GOLDSBOROUGH, 2016)

Nota: À esquerda, os autores testaram um modelo pequeno com uma única camada oculta e 200 unidades; à direita eles usam duas camadas com 650 unidades cada.

3 TRABALHOS CORRELATOS

Existem trabalhos de classificação de espécies de pragas, com características específicas, baseadas no tipo de espécie a ser identificada, bem como trabalhos em que foram usadas redes neurais profundas para a classificação de imagens em geral.

Dermatologist-level classification of skin cancer with deep neural networks

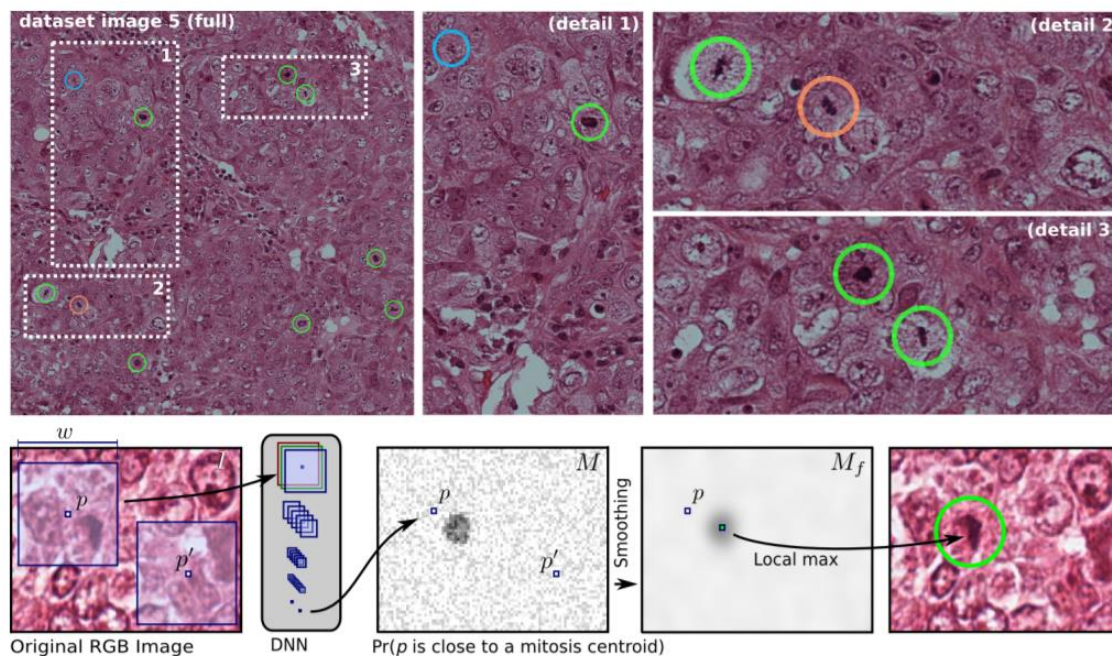
O câncer de pele é primariamente diagnosticado visualmente, por meio de uma triagem clínica inicial e potencialmente seguida de análise dermatoscópica, biópsia e exame histopatológico. A classificação automatizada de lesões cutâneas utilizando imagens é uma tarefa desafiadora devido à variabilidade de grãos finos no aparecimento de lesões cutâneas. Neste trabalho (ESTEVA, et. al., 2017) é demonstrado como foi feita a classificação de lesões cutâneas usando uma única CNN, treinada de ponta a ponta a partir de imagens, usando apenas pixels e rótulos de doenças como entradas. A CNN foi treinada usando um conjunto de dados de 129.450 imagens clínicas, consistindo de 2.032 doenças diferentes. O desempenho foi testado contra 21 dermatologistas certificados em biópsias, em imagens clínicas comprovadas por biópsia com dois casos críticos de uso de classificação binária: carcinomas de queratinócitos e queratoses seborreicas benignas; e melanomas malignos e nevos benignos. O primeiro caso representa a identificação dos cânceres mais comuns, o segundo representa a identificação do câncer de pele mais letal. A CNN atingiu desempenho parecido com todos os especialistas testados em ambas as tarefas, demonstrando assim que a CNN foi capaz de classificar o câncer de pele com um nível de competência comparável aos dermatologistas.

Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks

No trabalho de Cireşan (CIREŞAN, et al., 2013), os pesquisadores utilizaram redes neurais convolucionais para detectar mitose em imagens histológicas da mama. As redes foram treinadas para classificar cada pixel nas imagens, usando como contexto um patch centrado no pixel. A abordagem ganhou a competição de detecção de mitose ICPR 2012, superando outros concorrentes por uma margem significativa. O problema de classificação foi resolvido pela criação de um detector em imagens de treinamento com informações reais dadas sobre o centróide de cada mitose visível. Cada pixel é atribuído a uma das possíveis classes, mitose ou não-mitose. Para qualquer dado pixel p , a rede

neural prevê sua classe usando valores RGB brutos em uma janela de imagem quadrada centrado em p .

Figura 52- Esquema de classificação da rede neural profunda



Fonte: (CIREŞAN, et al., 2013).

Nota: Canto superior esquerdo: uma imagem correspondente a uma do conjunto de dados. A mitose detectada é circulado em verde (positivos verdadeiros) e vermelha (falsos positivos); Ciano denota mitose não detectada pela abordagem. Superior direito: detalhes em tamanho real de três áreas. Inferior: visão geral da abordagem.

Digital Automated Identification System (DAISY)

DAISY é um sistema genérico de correspondência de padrão que foi inicialmente desenvolvido pelo Museu Natural de História de Londres. Desenvolvido em meados de 1990 pelo Doutor Mark A. O'Neill, tem como objetivo principal para o desenvolvimento deste sistema a superação de um impedimento taxonômico e fornece um sistema que permita que não especialistas identifiquem organismos dentro de gêneros de artrópodes específicos usando uma combinação de morfologia e dados moleculares. O sistema DAISY classifica aranhas, grãos de pólen, e borboleta através da análise de componentes principais (ACP).

Baseado na ACP, DAISY identificou com precisão cinco espécies de vespas parasitas após ser treinado com conjuntos de dados de que continham de 8 a 10 asas por espécie (Weeks et al., 1997). Um resultado semelhante foi alcançado com 30 espécies de

maruim (Weeks et al., 1999). Essas identificações alcançaram mais de 95% de precisão para identificação de moscas, baseado em suas asas.

DAISY foi construído em um ambiente Linux usando modelo de rede neural artificial (ANN) e com isso seu aprendizado é dinâmico. Os modelos ANN são amplamente utilizados em reconhecimento de padrões.

Posteriormente uma nova versão do DAISY foi desenvolvida, denominada de DAISY II, essa versão tinha por objetivo abrandar a não-linearidade e problemas de escala da abordagem baseada em ACP. A nova versão foi construída usando um modelo de classificação vizinho mais próximo (NNC, do inglês *nearest-neighbour classification*). O NNC foi escolhido pelo seu simples, mas efetivo esquema de classificação. Apesar da fácil compreensão e execução da classificação baseada em NNC, algoritmos que foram originados por essa abordagem atingiriam uma precisão maior que 95% quando comparados a algoritmos de correspondência de padrões mais complicados (Lucas, 1997).

SPecies IDentified Automatically (SPIDA-web)

SPIDA-web é o trabalho da ecologista Kimberly N. Russell e do cientista Martin T. Desenvolvido no Museu Nacional Americano de História Natural em Nova York. O sistema usa redes neurais artificiais para identificar espécies. O sistema atribui a cada parte relevante do corpo uma identificação única.

No processo de aprendizagem do sistema, é necessário fotos de estruturas taxonomicamente relevantes da espécie, por meio de uma câmera comum montada no microscópio. Esse processo chamado transformação wavelet (semelhante à transformação de Fourier) reduz progressivamente a imagem para uma cópia mais simples de si mesma e preserva a informação, eliminando detalhes desnecessários, como cabelos ou espinhas. Essa tecnologia obtém resultados expressivos quando aplicada em problemas como reconhecimento de voz e face, e muitas vezes é usada por instituições financeiras e policiais.

Inicialmente, o SPIDA-web classifica apenas duas famílias de aranhas australianas (*Trochanteriidae* e *Prodidomidae*), juntamente com outro grupo de aranhas coletadas no Tennessee, Estados Unidos da América. Cada espécie tem sua própria rede neural e o sistema passa por elas até encontrar uma correspondência. Caso seja inserido

no sistema uma imagem da qual ele não foi treinado para reconhecer, o SPIDA-web irá incorporá-la como desconhecida.

Especialistas da área também contribuem com o aprendizado do sistema, por meio da identificação de espécies inseridas no banco de dados do sistema. Para cada novo grupo, os pesquisadores precisariam criar e treinar novas redes neurais artificiais, com isso, a dificuldade de implementação do sistema aumenta, devido que para cada novo grupo uma nova rede neural artificial deve ser criada, sendo descartada o possível reuso de redes neurais já existentes.

Automated Identification of Bee Species (ABIS)

Em tradução livre "Identificação automatizada de espécies de abelhas" (ABIS), o sistema foi desenvolvido a fim de identificar espécies de abelhas a partir de imagens de suas asas. As asas das abelhas mostram uma clara disposição das nervuras, dentro de uma pele transparente quando expostas à uma iluminação especial. A estrutura destas nervuras é geneticamente fixada e, portanto, pode ser utilizada na identificação de espécies. Podem ser extraídas cerca de 300 características à partir de cada imagem capturada. Essas características são então empregadas em uma nova abordagem de classificação baseada no conhecimento para identificar as espécies de abelhas. A análise e classificação da imagem são realizadas em uma hierarquia de operações consecutivas. Primeiro, o gênero da abelha é identificado. Posteriormente, o ABIS identifica os grupos de espécies. E no final do processo de análise e classificação, é utilizado uma nova análise discriminante não linear para realizar a identificação das espécies.

O resultado dessa classificação pode ser visualizado em termos de imagens de cluster ou dendrogramas. O equipamento necessário para o uso do ABIS é portátil e adequado para trabalhos no campo.

O ABIS foi utilizado na Alemanha, Brasil e nos EUA e apresentou resultados relevantes na identificação de abelhas. O sistema também pode ser aplicado a outros insetos, como vespas ou moscas.

Mid-level Image Representation for Fruit Fly Identification

As espécies de moscas-das-frutas do gênero *Anastrepha* são economicamente importantes na América, com aproximadamente 300 espécies conhecidas, onde existem 120 espécies registradas apenas no Brasil. No entanto, poucas espécies são consideradas pragas de importância por muitas agências reguladoras. Neste trabalho, foi proposto o uso de uma representação de imagem de nível médio (tipo de método de extração de características de imagem) para a identificação de insetos de três espécies do gênero *Anastrepha* usando diferentes descritores locais baseados em pontos-chave. Além disso, um sistema foi desenvolvido para identificar em tempo real as moscas-das-frutas.

As etapas de classificação de imagens são tipicamente compostas por três passos: extração de recursos visuais locais, onde informações são diretamente extraídas dos pixels da imagem; extração de recursos de nível médio, o que torna a representação mais geral, retirando informações que possam ser descartadas da imagem; e a última etapa é a classificação supervisionada, onde uma técnica de aprendizagem de máquina é empregada a fim de permitir a extração de um modelo geral a partir dos dados.

No que diz respeito a análise de imagens, a extração de recursos é o primeiro passo, seu objetivo é de extrair características visuais de certas regiões da imagem por meio de operações realizadas em casa pixel. De acordo com a área relativa dessas regiões, as características extraídas podem ser classificadas como global, onde um vetor de recursos é criado para representar o todo da imagem, ou local, onde são criados vários vetores de recursos por imagem, onde pequenas regiões da imagem são separadas e representadas separadamente.

A extração de recursos locais geralmente inclui dois tipos distintos que são: detecção de recursos e descrição da característica. A detecção consiste em encontrar um conjunto de pontos de interesse, já a descrição das características baseasse na obtenção de descritores locais dos recursos detectados.

A extração de recursos de nível médio visa a transformação local de descritores em uma representação de imagem global e mais rica de complexidade intermediária. Os passos para obter os recursos de nível médio podem ser divididos em duas etapas: codificação e agrupamento. Na etapa de codificação são quantificados os descritores locais de acordo com um dicionário visual de k palavras visuais, que são construídos agrupando um conjunto de descritores locais. A etapa de agrupamento agrega os

códigos obtidos em um único vetor de características. A etapa de codificação associa os descritores locais ao elemento mais próximo do dicionário visual e o agrupamento leva a média desses códigos. A grosso modo, o objetivo do agrupamento é resumir as informações contidas nos descritores em um único vetor de características, preservando assim informações importantes e descartando detalhes irrelevantes.

Análises foram feitas a fim de avaliar a qualidade dos pontos-chave detectados por meio de extratores de características. Os extratores foram analisados com base em seu desempenho ao avaliar imagens de asas das moscas. Posteriormente uma análise de eficácia entre oito descritores locais e nove técnicas de aprendizagem foram realizadas para verificar o comportamento das tuplas (extrator de características combinado com a técnica de aprendizagem) na tarefa de identificação da mosca da fruta, resultando assim na conclusão de que o BREVE e o F-ETERDAL conseguiram os melhores resultados de precisão média entre todos os descritores locais testados. Já com relação as técnicas de aprendizagem, a *Multilayer Perceptron* (MLP) e *Support Vector Machine* (SVM) se destacaram como as sendo as melhores técnicas de aprendizagem com valores de precisão média mais altos e menor intervalo de confiança. Como trabalho futuro, serão realizados experimentos com outros extratores, espécies e técnicas de aprendizagem. Além disso, um sistema móvel deve ser desenvolvido para ajudar especialistas da área de biologia em seus trabalhos de campo.

DrawWing

Ferramenta que permite a identificação automática das junções das veias das asas de insetos, baseada no padrão específico de cada espécie. Uma imagem de uma asa de inseto é usada a fim de determinar o contorno das veias. Os pontos característicos são muitas vezes junções de veia ou extremidades de veia. A distribuição da veia é obtida por meio de um algoritmo de desbaste. É aplicada também as curvas de Bezier, onde são adaptadas tanto ao contorno da asa como ao contorno da veia. As curvas são salvas em um arquivo *postscript*. As coordenadas de junção da veia são salvas em outro arquivo em formato de texto. O *DrawWing* fornece automaticamente uma descrição numérica de uma asa de inseto. Converte uma imagem de uma asa de inseto em uma lista de coordenadas de junções de veia e um diagrama de asa que pode ser usado como ilustração. Atualmente, o *DrawWing* permite a identificação de abelhas baseado em suas asas, em trabalhos

futuros o sistema irá permitir o reconhecimento de qualquer inseto, baseado também em suas asas.

O *DrawWing* pode extrair características de asas de insetos automaticamente, isso faz com que o sistema seja capaz de identificar insetos discriminando um grande número de espécies. Atualmente, o *DrawWing* não executa a classificação automaticamente. Contudo o sistema é gratuito e de fácil acesso para uma ampla gama de usuários potenciais. O *DrawWing* é o único programa de computador para a extração automática de características importantes de asas de insetos que são distribuídas gratuitamente em conjunto com seu código fonte.

DAIIS

Sistema que também se baseia no contorno da asa de insetos para identificá-los. O sistema DAIIS inclui duas funções principais: digitalização de tópicos e transformação de Fourier elíptica e treinamento do modelo classificador para reconhecimento de padrões de máquinas de vetor de suporte e validação de modelos. A fim de demonstrar a utilidade deste programa, amostras de 120 corujas foram divididas em conjuntos de treinamento e testes. Feito isto, a amostra foi classificada em sete espécies usando o DAIIS. Em cinco experimentos repetidos, a precisão média para a identificação de cada espécie variou de 90% a 98%. A precisão aumentou para 99% quando as amostras foram primeiro divididas em dois grupos com base nas características dos seus olhos compostos.

O DAIIS foi desenvolvido com o intuito de ser uma ferramenta simples para ajudar usuários que não possuem um amplo conhecimento de algoritmos e programação para desenvolver seus próprios sistemas de identificação automática com base em contornos de partes do corpo. Sendo assim, basta simplesmente baixar o pacote de software e copiar as imagens de destaque para as pastas "doTrain" e "doPre" para treinar o modelo de identificação e posteriormente realizar a identificação propriamente dita, respectivamente. Com isso, os usuários não precisam se preocupar em projetar o algoritmo, e desta forma, podem se concentrar em sua área de especialização. O DAIIS foi a primeira ferramenta para o desenvolvimento de sistemas de identificação automática para qualquer espécie usando SVMs e coeficientes de EF dos contornos das partes do corpo.

Com o intuito de reduzir a taxa de erro ao classificar os insetos, foi desenvolvido um conjunto de regras para a captura e processamento de imagens. Foram determinados 30 harmônicos que produziram 120 coeficientes a fim de reconstruir um esboço preciso com a transformação elíptica de Fourier. Foram realizados também transformações desses coeficientes com o objetivo de reduzir o impacto de variância no tamanho das imagens, rotação e ponto de partida para traçar um esboço, dessa forma a precisão na identificação obteve uma melhora significativa.

Podemos observar que todos esses trabalhos citados acima possuem dois processos principais: extração de características taxonômicas e o desenvolvimento de um modelo classificador para posterior identificação. Os trabalhos citados demonstram a importância do desenvolvimento de sistemas cujo objetivo seja de classificar pragas, a fim de auxiliar tanto profissionais da área taxonômica, de agronomia, ou pessoas que não possuam conhecimento especializado no que diz respeito a classificação de pragas.

Contudo, não se tem conhecimento na literatura de trabalhos cujo objetivo seja de classificação de lagartas, mais precisamente lagartas que atacam a cultura da soja.

Desta forma destaca-se a necessidade do desenvolvimento de tal sistema, do qual possa auxiliar a classificação de lagartas, devido à importância econômica causada por ataques desse tipo de praga, bem como a importância de espalhar o conhecimento sobre a classificação das mesmas, facilitando o acesso a informação e descentralizando o conhecimento necessário para este tipo de prática de especialistas.

Detection of insects in bulk wheat samples with machine vision

Infestação de grãos é um problema sério que causa prejuízos financeiros e riscos para a saúde. Existe uma variedade de fatores que afetam o valor do trigo, a presença de insetos vivos é um deles. De acordo com os padrões oficiais de classificação de grãos dos Estados Unidos de 1997, a designação de grau de “infestada” é dada quando “A amostra representativa contém dois ou mais gorgulhos vivos ou um gorgulho vivo e um ou mais insetos vivos prejudiciais a grãos armazenados, ou dois ou mais insetos vivos prejudiciais a grãos armazenados”.

Depois que a amostra de grãos é feita, o material é levado ao laboratório e dividido em amostras, uma amostra de trabalho, uma amostra de arquivo e uma amostra de

umidade. A amostra de trabalho é usada pelo avaliador para determinar a sua condição. Os primeiros fatores que o avaliador verifica são odor, infestação e outras condições incomuns. Toda a amostra de trabalho é verificada visualmente afim de encontrar insetos vivos, que são separados do grão com peneiras próprias. Insetos e outros materiais finos são coletados em uma espécie de peneira que fica abaixo da peneira quando o grão é agitado sobre a peneira. Feito isso, os insetos encontrados são contados e identificados. Isso pode ser um procedimento trabalhoso, além disso, erros humanos são bastante comuns, o que acarreta na imprecisão do processo.

Existem diversas técnicas que são vistas como possíveis alternativas do método tradicional, sendo uma delas a Visão de Máquina. Para fins de controle e inspeção a metodologia de reconhecimento de padrões em combinação com visão de máquina é utilizada em muitas áreas da indústria e agricultura. Existem estudos sobre o uso de técnicas de imagem para morfometria descritiva de insetos individuais, como cupins, abelhas e mosquitos, mas até então não existiam estudos relatados sobre amostras. Podemos citar também técnicas de imagem usando fotomultiplicadores e técnicas de reconhecimento de padrões usados para reconhecimento de vários insetos, onde o foco foi a aplicação em campo de algodão. Esse estudo analisou objetos individuais no campo de visão e obteve uma taxa de reconhecimento de aproximadamente 85%.

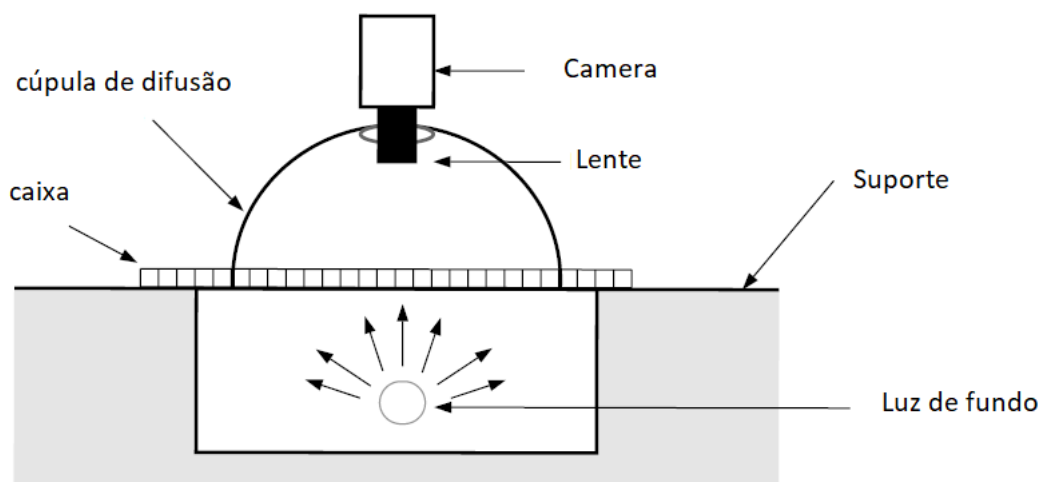
A visão de máquina é uma técnica de reconhecimento de padrões atrativa para a abordagem de inspeção de qualidade de grãos, onde o algoritmo desenvolvido pode fazer a contagem de insetos, exibir insetos detectados para confirmação visual e arquivamento de amostras, se necessário.

Nesse estudo, o autor teve como objetivo explorar a possibilidade de desenvolver um método para detectar insetos inteiros ou partes do corpo nas amostras a granel de grãos e docagem, usando visão de máquina. O principal objetivo do estudo foi determinar coordenadas de uma sub-imagem, onde elementos da imagem seriam classificados como sendo insetos ou não insetos. A procura pelas coordenadas foi baseada em medições multiespectrais, extraídas de imagens em vermelho, verde e azul para a mesma amostra. Para reduzir o tempo computacional, o autor usou sub-imagens ao invés de pixels individuais. Uma das preocupações do estudo foi determinar qual o tamanho da sub-imagem deveria ser usado para executar um programa de forma eficiente, afim de reduzir a quantidade de informação para processar. A determinação dos limites do corpo de insetos era outro objetivo desafiador, considerando diferentes ângulos de insetos ou partes de insetos em um tamanho de amostra e sub-imagem. A semelhança na cor dos insetos a

alguns componentes da docagem tornou a tarefa ainda mais difícil. Além de que partículas de poeira obscurecem o tecido dos insetos e modifica a resposta de dispersão multiespectral.

O material usado foi amostras de grãos de trigo que se tornou infestado com várias espécies de insetos de produtos armazenados. Sendo elas *Rhyzopertha dominica*, *Cryptolestes ferrugineus*, *Cephalonomia waterstoni*, *Choetospila elegans*, *Oryzophilus surinamensis* e *Triboleum castaneum*. Um amostrador de grãos pneumáticos foi usado para obter amostras de grãos de 35,2 m³ de trigo. As amostras eram uma mistura de grãos de trigo e docagem, incluindo algumas sementes de grama. As amostras de 3 kg foram passadas sobre uma peneira inclinada (89 × 43 cm, 1,6 mm de abertura) onde os grãos foram separados de insetos e docagem. Alguns grãos de trigo foram adicionados de volta ao material peneirado antes da aquisição de imagens para determinar se o trigo poderia ser distinguido das partes de insetos. As imagens foram capturadas por meio de hardware e software da Kontron Electronic GmbH. O software IBAS 2.5 foi instalado em um computador 486 DX de 33 MHz com sistema operacional DOS, com 16 Mb de memória. O processamento e visualização de dados foi feito usando uma estação de trabalho 75 MHz SPARC LX. As imagens foram obtidas com uma câmera colorida Progress 3000 montada em um *photostand* da Bencher. A câmera colorida Progress 3000 foi calibrada para resolução de 1488 × 1180 pixels, uma das seis calibrações possíveis. O formato das imagens adquiridas foi de 512 pixels, com uma proporção de pixel de 1,06. O campo de visão foi de 5,4 mm x 5,3 mm. Os valores de calibração foram escala x: $1,05 \times 10^{-2}$ mm = 1 pixel; y: $1,03 \times 10^{-2}$ mm = 1 pixel. As amostras foram colocadas em uma caixa. A profundidade da amostra foi de aproximadamente 5 mm e o fundo da caixa estava coberto. A calibração de cores da câmera Progress foi programado para amostragem de pixels brancos e pretos. Um ou mais insetos estavam presentes no campo de visão para a realização da amostragem. As amostras foram agitadas para expor os insetos.

Figura 24 – Ilustração do sistema de captura de imagens



Fonte: Adaptado de: (ZAYAS; FLINN, 1998).

Por causa da semelhança de cor e brilho, alguns componentes da docagem foram difíceis de diferenciar dos insetos, como sementes de grama. A posição do inseto (ventral, dorsal e lado) e partículas agarradas ao inseto afetam o sucesso de reconhecimento. Para a melhor qualidade de imagem, considerando os problemas acima mencionados foi adotada iluminação de fundo, usando lâmpada de halogéneo de 250 W. Para reduzir a reflexão, uma cúpula com tinta branca fosca no interior foi colocada sobre a amostra. A câmera Progress 3000 foi colocada acima da cúpula com uma lente Nikkor 50 mm dentro a cúpula. Imagens de setenta amostras foram armazenadas para análise futura.

As cores vermelho, verde e azul foram extraídas de histogramas para cada uma das sub-imagens bem com as características de matiz, saturação e luminosidade. Trinta características dos histogramas de sub-imagem incluíam a média, desvio padrão, assimetria, e outras características derivadas. A análise multivariada foi utilizada para classificar as observações, que eram sub-imagens de insetos, sementes de grama e outros para duas e três categorias de classe. A análise discriminante foi executada usando um programa escrito pelo próprio autor e procedimento SAS (SAS, 1991) para verificação dos resultados da análise discriminante. Nas etapas de treinamento e teste, o programa computou a probabilidade de cada sub-imagem pertencer a uma determinada classe.

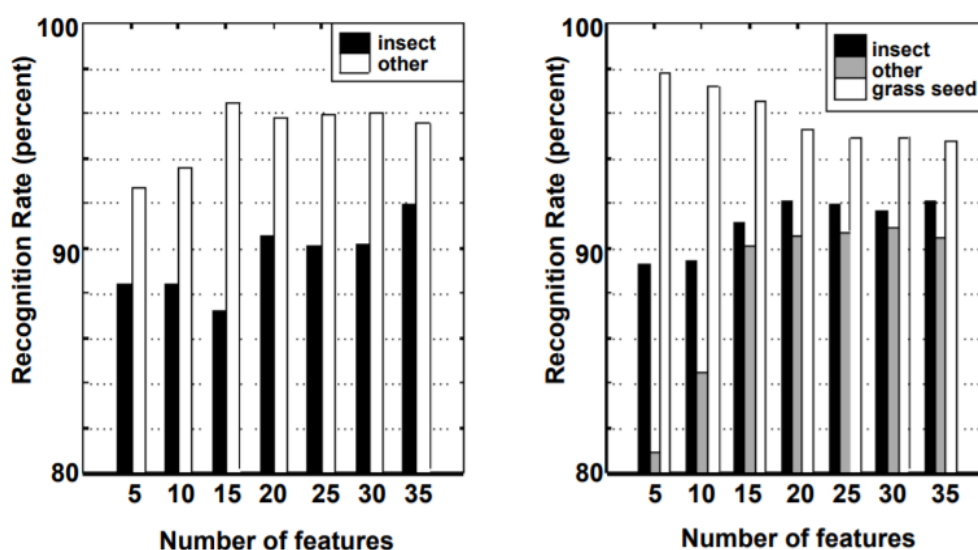
As classes foram definidas em: “Inseto e não inseto” (para análise de duas classes distintas) e “Insetos, sementes de grama e outros” (para análise de três classes distintas). Sementes de grama foram escolhidos de outros componentes da docagem, porque eles

eram muito parecidos em cor aos insetos. "Outras" incluiu todos os tipos de docagem e áreas sombreadas entre objetos no campo de visão.

Os experimentos de treinamento e testes foram feitos de duas maneiras diferentes. Na primeira abordagem, treinamento e conjuntos de dados de teste foram criados pela seleção interativa de sub-imagens representativas de três classes: insetos (3932), sementes de gramíneas (2432) e outras (2244). O pool de sub-imagens foi aleatoriamente dividido ao meio para conjuntos de treinamento e teste para cada classe. Análise discriminante e canônica foram executados por meio do pacote estatístico SAS (SAS,1991) e um programa interno foi executado para computar funções lineares e quadráticas com diferentes características dos modelos. A análise discriminante também foi executada em conjunto com análise canônica; funções canônicas foram computadas primeiro, e os resultados foram usados como uma entrada para o cálculo de funções discriminantes. O procedimento STEPDISC do SAS foi usado para escolher o número de recursos no modelo.

Como resultados, a categoria de “duas classes” obteve melhor desempenho quando 35 características foram usadas, onde o modelo obteve 91,3% de taxa de acerto para insetos e 96,3% para outros. Já a categoria “três classes” teve um melhor desempenho quando usado 25 características, alcançando assim 91,3% para insetos, 91,2% para outros e 94,7% para sementes.

Figura 25 - Gráficos de barras de taxa de reconhecimento correta para duas e três classes para modelos de características



Fonte: (ZAYAS; FLINN, 1998).

Ao avaliar os trabalhos correlatos aqui citados, pode-se observar a importância dos métodos de classificação com o intuito de classificar insetos, seja para controle ou não dos mesmos. Além disso, por meio de sistemas autômatos pode-se popularizar o processo de classificação, onde qualquer pessoa que tenha interesse nessa área possa usar esses sistemas afim de realizar a classificação, sem a necessidade de conhecimento prévio. Outros sistemas podem usufruir dessa tecnologia, por exemplo, um sistema computacional embarcado em um drone, pode realizar o levantamento do número de pragas em uma lavoura.

4 METODOLOGIA

Com o intuito de avaliar a metodologia que melhor se encaixa na resolução do problema de classificação das pragas *Anticarsia Gemmatalis*, *Helicoverpa Armigera* e *Spodoptera Frugiperda*, foram estudados inicialmente artigos e materiais bibliográficos que tratam do tema processamento digital de imagens e suas aplicações. A aplicação de redes neurais convolucionais foi escolhida, devido a que se tornaram a base da maioria das soluções de visão computacional de última geração para uma grande variedade de tarefas. Desde 2014, as redes de convolução profundas começaram a se tornar *mainstream*, gerando ganhos substanciais em vários benchmarks (SZEGEDY, 2016).

As imagens utilizadas neste trabalho foram coletadas a partir da internet e de campo, e não foram submetidas a quaisquer pré-processamento, como alteração do espaço de cores, redimensionamento, etc. A não utilização da etapa de pré-processamento se justifica pela não tratativa das imagens, para que assim as classificações não sejam de alguma forma beneficiadas pela etapa de pré-processamento. As imagens foram colhidas e utilizadas sem padronização de tamanhos, não foram submetidas a filtros ou a qualquer outra forma de melhoria.

Com relação à classificação, é sabido que as formas geométricas dos objetos são um meio bastante confiável para que possamos realizar a identificação do mesmo. Podemos reconhecer objetos independentemente de sua coloração, apenas nos baseando em suas características de forma. Entretanto, no presente estudo, devemos levar em consideração as cores das pragas, porque as cores são um fator determinante na classificação das mesmas já que se levarmos apenas em conta a forma das lagartas, não iremos obter um resultado satisfatório, devido a que o formato das lagartas é praticamente

igual, independente da sua espécie. Além disso, as redes neurais convolucionais tem a característica de realizar a extração automática de características de imagem, sem que isso seja feito de maneira manual.

Foi implementado um sistema do tipo cliente-servidor, onde a parte do servidor fica à cargo da classificação das imagens enviadas pelo cliente. O sistema foi desenvolvido a fim de servir como meio de validação do método de classificação.

4.1 AQUISIÇÃO DE IMAGENS

As imagens das pragas *Anticarsia Gemmatalis*, *Helicoverpa Armigera* e *Spodoptera Frugiperda* foram coletadas a partir da internet e de campo. Não há uma base de dados com imagens dessas pragas de conhecimento público até o presente momento e também a aquisição de imagens de pragas em campo se torna um problema tanto quanto desafiador, pois é necessário capturar um número considerável de imagens para que a classificação seja feita de maneira satisfatória, esses fatores dificultaram o processo de aquisição de imagens. Com isso, a relação do número de imagens capturadas por praga é descrita abaixo:

Tabela 3 - Número de imagens coletadas por espécie

Espécie	Nº de imagens coletadas
<i>Anticarsia Gemmatalis</i>	100
<i>Helicoverpa armigera</i>	125
<i>Spodoptera Frugiperda</i>	99

Fonte: O autor.

No total foram coletadas 324 imagens, com uma média de 108 imagens por espécie de praga. As imagens foram aferidas por um profissional especialista na área. Não foram aplicadas nenhuma espécie de tratamento nas imagens, seja de redução de ruídos, filtros ou qualquer outra espécie de melhoria da imagem. Isso se dá ao objetivo de poder utilizar imagens conforme foram capturadas, independentemente do seu tamanho ou câmera utilizada. Os únicos requisitos obrigatórios foram o formato das imagens, que é *Joint Photographic Experts Group* (mais conhecido como JPEG ou JPG), e as imagens

também precisam ser coloridas, pois as cores serão usadas na classificação das pragas. Além disso, a rede CNN possui uma característica bastante relevante, pode aprender/extrair as características por ela mesma, sem a necessidade de que seja realizada a extração das características anteriormente ao processo de aprendizagem. Com isso, a rede CNN é capaz de extrair as características por ela mesmo, e posteriormente usar essas mesmas características para realizar a classificação.

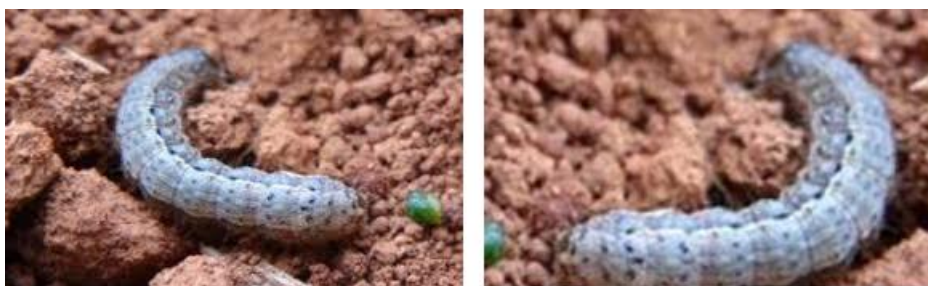
Pelo fato de o número de imagens ser bastante limitado, foram criadas 676 novas imagens aleatórias, a partir das 324 imagens, totalizando assim 1000 imagens, onde transformações como rotação, inversão de lados e aproximação foram aplicadas, as imagens então foram rotuladas e agrupadas conforme a espécie de praga.

Figura 53 - Script de criação de novas imagens a partir de transformações das imagens originais

```
1 import Augmentor
2 p = Augmentor.Pipeline("Imagens Originais")
3 p.rotate(probability=1, max_left_rotation=5, max_right_rotation=5)
4 p.flip_left_right(probability=0.5)
5 p.zoom_random(probability=0.5, percentage_area=0.8)
6 p.sample(676)
```

Fonte: O autor.

Figura 54 - (a) Imagem original. (b) Imagem após receber as transformações de rotação e aproximação



(a)

(b)

Fonte: O autor.

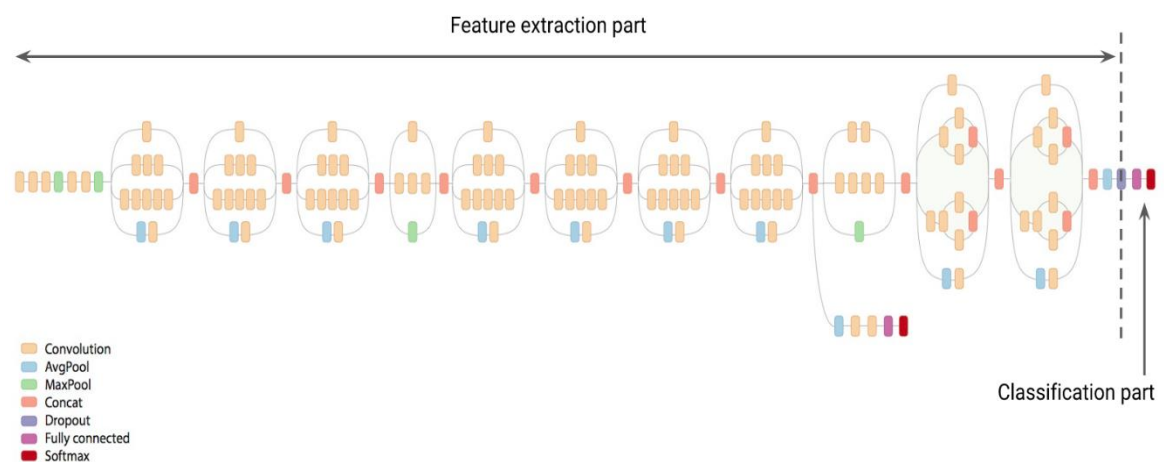
4.2 SISTEMA COMPUTACIONAL

O sistema de processamento que foi desenvolvido para classificar as pragas *Anticarsia Gemmatalis*, *Helicoverpa Armigera* e *Spodoptera Frugiperda* foi implementado na linguagem de programação Python, versão 2.7. Os autores Liu et al. (2015) utilizaram a linguagem de programação Matlab para o desenvolvimento do sistema de processamento digital de imagens.

A plataforma de desenvolvimento adotada foi o Visual Studio Code 1.25.1 e o computador utilizado foi um notebook da marca Dell, com processador Intel® Core™ i7 – 5500U 2.40 GHz e 16GB de memória RAM, com sistema operacional Windows 10 64-bits.

A plataforma de aprendizagem profunda utilizada foi o Tensorflow 1.6, para treinar, validar e testar a rede neural, o modelo Inception V3 foi usado, ele foi previamente treinado em outro problema semelhante, a aprendizagem profunda quando iniciada do zero pode levar dias para ser concluída, agora reutilizando modelos por meio do aprendizado de transferência, pode-se fazer em pouco tempo. O modelo foi previamente treinado no conjunto de dados *ImageNet Large Visual Recognition Challenge (ILSVRC)*, onde conta com mais de 1.000 classes diferentes, dos mais diversos segmentos e contém mais de 14 milhões de imagens em seu dataset. A arquitetura da rede neural convolucional Inception V3 (CHRISTIAN ET AL., 2015) foi utilizada e na classificação da base ILSVRC14 e atingiu o estado-da-arte para a classificação e detecção. Abaixo temos uma ilustração da arquitetura da CNN Inception em sua 3ª versão.

Figura 55 - Arquitetura da CNN Inception-V3



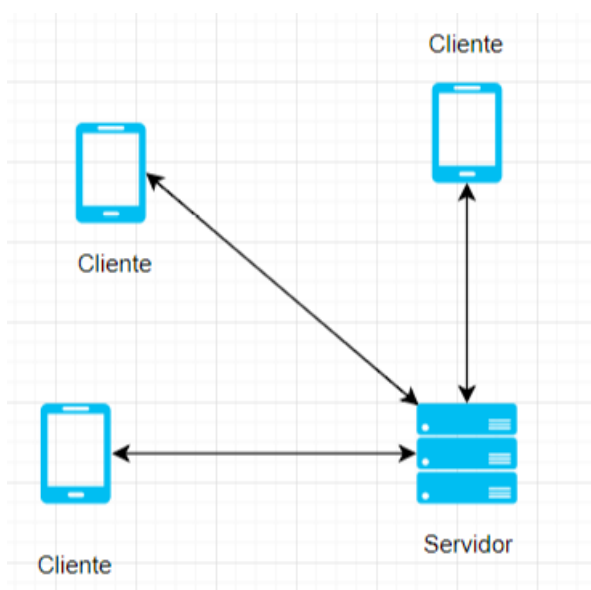
Fonte: Image Classification Transfer Learning with Inception v3. Disponível em: <<https://codelabs.developers.google.com/codelabs/cpb102-xf-learning/index.html?index=.%2F.%2Findex#0>>. Acesso em 20 de mar. 2018.

A camada final de classificação da rede neural foi removida e recriada a partir dos dados contidos na base de imagens das lagartas. Os parâmetros da rede foram finamente ajustados. Durante o treinamento as imagens foram redimensionadas para 299×299 pixels, tornando-as compatíveis com as dimensões originais das imagens da base ImageNet, usada na rede Inception v3. A CNN é treinada usando *backpropagation* e todas as camadas da rede são refinadas usando a mesma taxa de aprendizagem global de 0,01 que difere do trabalho de Esteva (ESTEVA et. al., 2017), onde a taxa foi ajustada para 0,001.

Depois de desenvolvida a parte de classificação, foi então criado um servidor do qual irá processar as requisições advindas de clientes, essas requisições são atendidas por meio do protocolo http sigla em inglês de *Hypertext Transfer Protocol* (em português, Protocolo de Transferência de Hipertexto). Sendo assim, para usar o serviço de classificação de imagens, basta que a aplicação cliente faça uma requisição HTTP do tipo POST para a URL do servidor. Essa requisição deve enviar a imagem no formato Base64, que é um método que codifica os dados para serem transferidos na internet. Com isso, é simples fazer com que diversos tipos de softwares utilizem o serviço de classificação de imagens.

Concluído o desenvolvimento do servidor, foi então criado um aplicativo do qual irá consumir o serviço fornecido pelo servidor.

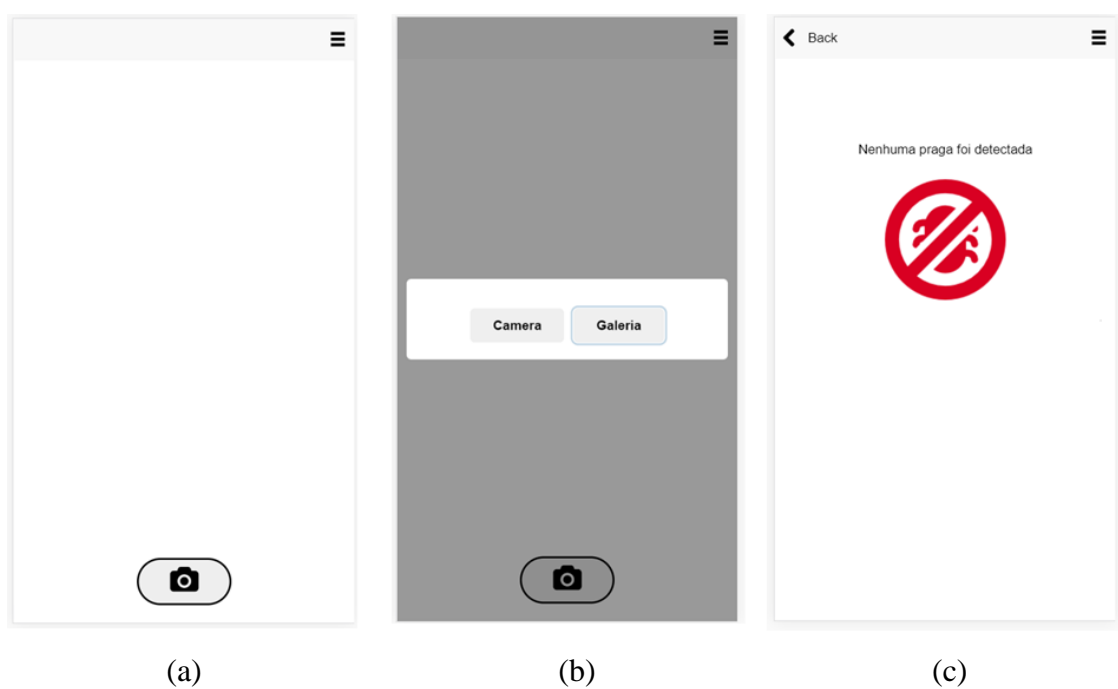
Figura 56 - Arquitetura de sistema cliente-servidor



Fonte: O autor.

O aplicativo foi desenvolvido com o intuito de ser fácil de usar, intuitivo, e que não demandasse recursos em demasia do aparelho móvel em que fosse executado, para que assim ele também fosse compatível com aparelhos móveis cujo recursos são bastante limitados.

Figura 57 – (a) Tela inicial do aplicativo. (b) Tela de seleção de origem da imagem a ser classificada. (c) Tela mostrada em caso de nenhuma praga encontrada na imagem.



Fonte: O autor.

Figura 58 - Tela mostrada em caso de classificação realizada com sucesso



Fonte: O autor.

Conforme a figura 56 mostra, em caso de classificação bem-sucedida, uma tela é mostrada ao usuário, contendo imagem da praga, e informações sobre a mesma. Dessa forma, o usuário obtém informações das quais o auxiliam no conhecimento da praga em questão, e com isso, tomar decisões baseadas nessas informações.

5 RESULTADOS E DISCUSSÃO

A rede neural convolucional obteve uma taxa de acerto de 92.5%, a porcentagem de imagens reservadas para a validação foi de 25%, essas imagens não foram utilizadas no processo de treinamento, seu uso foi restrito apenas para a etapa de validação. A entropia cruzada ficou em 0.052433 e a precisão do treinamento ficou em 100%. A rede executou 4.000 etapas de treinamento, onde cada etapa selecionou aleatoriamente 10 imagens do conjunto de treinamento.

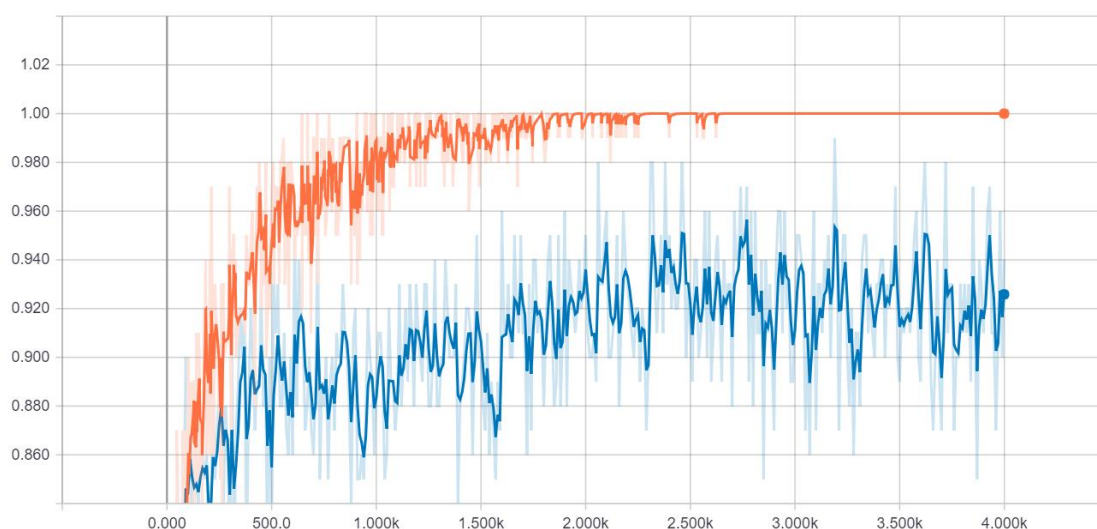
Tabela 4 - Resultados obtidos pela CNN

Etapa	Resultado
Entropia cruzada	0.052433
Precisão dos testes	100%
Precisão da validação	92.5%

Fonte: O autor.

Abaixo temos a ilustração do resultado do treinamento do presente trabalho:

Figura 59- Gráfico de assertividade do modelo treinado



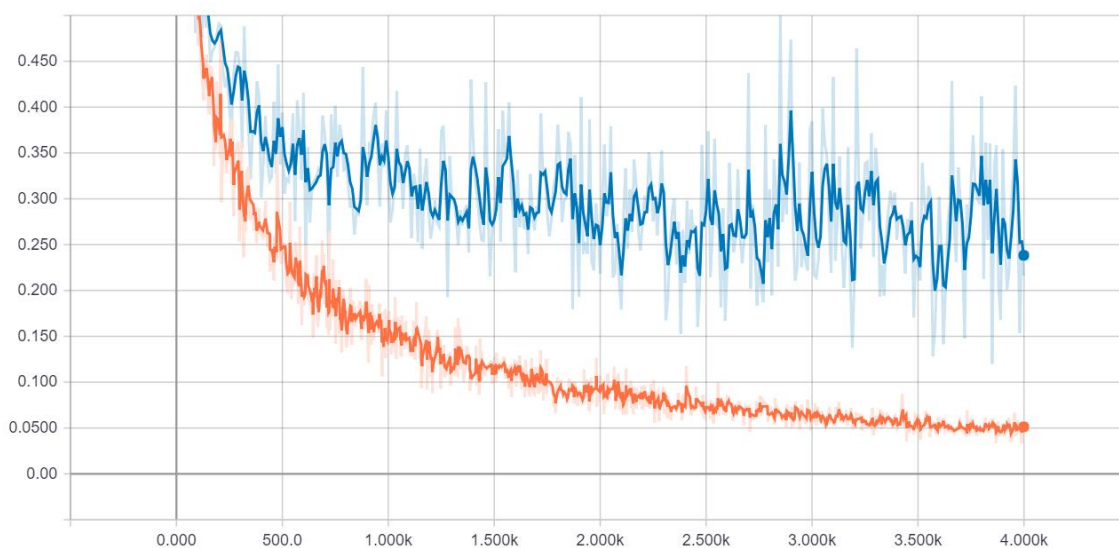
Fonte: O autor.

No eixo X, encontramos o número de passos de treinamento, 4.000 no total. No eixo Y encontra-se o grau de precisão do modelo. Em azul temos os valores correspondentes à validação e em laranja, temos os valores inerentes ao treinamento. A precisão do teste ficou em 100% e a índice de assertividade ficou em 92.5%

Para medirmos o real desempenho de uma rede neural, devemos testá-la com um conjunto de dados dos quais não foram usados para o treinamento. Para tanto, usamos a precisão de validação, caso o valor da precisão do treinamento for elevado e o valor da precisão de validação permanecer baixo, isso significa que a rede está sendo super adaptada e com isso memorizando recursos específicos nas imagens de treinamento que não são relevantes para classificar as imagens de maneira mais abrangente.

Podemos observar na imagem abaixo a evolução da entropia cruzada durante o treinamento do modelo:

Figura 60 - Evolução da entropia cruzada



Fonte: O autor.

Ao iniciar o treinamento da rede neural, a entropia cruzada tem seu valor inicial superior a 0.450, no decorrer dos passos de treinamento, fica claro a diminuição da entropia cruzada, finalizando o processo de treinamento do modelo em 0.052433.

O treinamento deve tornar o valor da entropia cruzada o mais próximo de zero, para que assim a rede neural esteja de fato aprendendo, a perda da entropia cruzada deve tender para baixo, ignorando o ruído de curto prazo.

No presente treinamento, foram executadas 4.000 etapas de treinamento. Cada etapa escolhe 10 imagens aleatoriamente do conjunto de treinamento, encontra seus *bottlenecks* no cache e as alimenta na camada final para obter previsões. Essas previsões são então comparadas com os rótulos reais para atualizar os pesos da camada final por meio de um processo de retropropagação.

Ao decorrer das etapas de treinamento, a precisão do modelo tende a aumentar. Uma avaliação final é executada ao final das etapas de treinamento, em um conjunto de imagens que ainda não foram usadas no treinamento e nem na avaliação. Dessa forma, podemos executar a tarefa de classificação em imagens que ainda não foram analisadas pelo modelo, tendo assim uma perspectiva melhor de como o modelo irá se comportar. Abaixo, vejamos a matriz de confusão dos testes executados nas 250 imagens reservadas para esse processo.

Tabela 5 - Matriz de confusão do resultado dos testes

	<i>Anticarsia gemmatalis</i>	<i>Helicoverpa armigera</i>	<i>Spodoptera cosmioides</i>
<i>Anticarsia gemmatalis</i>	75	6	3
<i>Helicoverpa armigera</i>	4	77	2
<i>Spodoptera cosmioides</i>	4	0	79

Fonte: O autor.

Ao avaliarmos a matriz de confusão, podemos observar que o erro mais recorrente foi relativo a classificação da praga *Anticarsia gemmatalis*, onde por 6 vezes o algoritmo classificou erroneamente como *Helicoverpa armigera* e 3 vezes como *Spodoptera cosmioides*. A espécie *Anticarsia gemmatalis* foi a que possuiu menor índice de acerto quando comparada com as outras duas espécies, obtendo 89,28% de assertividade.

A espécie *Helicoverpa armigera* obteve índice de assertividade de 92,77%, onde por 6 vezes foi classificada erroneamente, 4 delas como *Anticarsia gemmatalis* e em 2 ocasiões como *Spodoptera cosmioides*.

Por fim, a espécie *Spodoptera cosmioides* obteve o maior índice de assertividade dentre as espécies classificadas, com um total de 95,18% de assertividade. Essa espécie foi classificada corretamente em 79 imagens, e não obteve a mesma performance em 4 ocasiões, onde foi considerada como *Anticarsia gemmatalis*.

Tabela 6 - Ranking de assertividade por espécie

Espécie	Índice de Assertividade
<i>Spodoptera cosmioides</i>	95,18%
<i>Helicoverpa armigera</i>	92,77%
<i>Anticarsia gemmatalis</i>	89,28%

Fonte: O autor.

A espécie de praga que por mais vezes foi classificada erroneamente foi a *Anticarsia gemmatalis*, fato que ocorreu em 8 ocasiões. Em segundo lugar, a espécie *Spodoptera cosmioides* foi classificada erroneamente por 5 vezes, e por 6 vezes a espécie *Helicoverpa armígera* foi erroneamente considerada como a praga correta. Dessa forma, ao levarmos em consideração que foram realizados 250 testes, cada um em uma imagem da base de dados reservada para testes, podemos considerar que a somatória das classificações incorretas foi de 19. Esse número corresponde a cerca de 7,6% do total de imagens reservadas para teste.

Tabela 7 - Porcentagem de classificação incorreta por número de imagens por espécie

Espécie	Número de Imagens	Erros	Porcentagem
<i>Spodoptera cosmioides</i>	83	5	6%
<i>Helicoverpa armigera</i>	83	6	7,2%
<i>Anticarsia gemmatalis</i>	84	8	9,5%

Fonte: O autor.

Tabela 8 - Porcentagem de classificação correta por número de imagens por espécie

Espécie	Número de Imagens	Acertos	Porcentagem
<i>Spodoptera cosmioides</i>	83	79	95,18%
<i>Helicoverpa armigera</i>	83	77	92,77%
<i>Anticarsia gemmatalis</i>	84	75	89,28%

Fonte: O autor.

Ao avaliarmos as imagens em que foram classificadas erroneamente, podemos notar a semelhança de cores presentes nas imagens. Abaixo, vejamos uma imagem de *Anticarsia gemmatalis* que foi classificada erroneamente como *Helicoverpa armigera*:

Figura 61 - (a) Imagem de *Anticarsia gemmatalis* que foi classificada erroneamente como *Helicoverpa armigera*. (b) *Helicoverpa armigera*



Ambas as imagens possuem a cor verde predominante, e além disso, podemos notar uma coloração mais esbranquiçada, que forma uma lista na lateral de ambas as pragas. A imagem (b) foi usada durante o processo de treinamento da rede neural, e a imagem (a) foi usada durante o processo de teste. É compreensível que a rede neural tenha se enganado na classificação da imagem (a), visto que as imagens usadas no treinamento possuem características parecidas com a imagem (a). Dessa forma, esse problema pode ser reduzido ao utilizarmos imagens onde as características inerentes as espécies a serem ensinadas a rede neural sejam bastante diferentes. Essa solução pode ser facilmente adotada em casos em que as classes são bastante distintas, por exemplo no caso de um automóvel e um avião, já que ambos os objetos possuem formas bastante marcantes e que dificilmente possuem coloração e textura igual. No contexto das três pragas citadas no presente trabalho, podemos notar que as pragas possuem características bastante parecidas, como forma, cor e textura. A característica mais distinta nesse caso é a coloração, onde cada praga possui coloração distinta das outras classes, contudo é normal também que as pragas adquiram a coloração verde, essa mudança é comum e com isso, dificulta o reconhecimento das pragas, já que sua coloração passa a ser parecida com outras.

Figura 62 - Outro exemplo de classificação errônea. (a) *Anticarsia gemmatalis* classificada como *Helicoverpa armigera*. (b) *Helicoverpa armigera* real



Fonte: O autor.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um método para classificação de pragas das espécies *Anticarsia Gemmatalis*, *Helicoverpa armigera* e *Spodoptera Cosmioides*, onde uma rede neural convolucional foi utilizada. A rede neural é baseada na arquitetura Inception V3, que foi previamente treinada usando a base *ImageNet Large Scale Visual Recognition Challenge*, com mais de 1.4 milhões de imagens, divididas em 1.000 categorias. A camada final de classificação foi removida e então uma nova camada final de classificação foi criada a partir do treinamento usando a base de dados de pragas, usando transferência de conhecimento. A base de dados foi criada a partir de imagens das lagartas, onde 324 imagens foram usadas na geração de 676 imagens, onde foram aplicadas técnicas de rotação, inversão de lados, e aproximação, totalizando assim 1000 imagens. As formas de treinamento e validação da rede neural foi de 75% das imagens usadas para treinamento e os 25% restantes usados na validação, a separação de imagens usadas para treinamento e validação foi feita aleatoriamente. A rede neural obteve 92,5% de taxa de assertividade, o que configura um excelente resultado, já que não foram aplicadas quaisquer técnicas de pré-processamento nas imagens, e também da base de imagens não conter um número elevado de amostras, a rede neural convolucional obteve um índice de acerto considerável, o que confirma a superioridade da aplicação desse tipo de tecnologia quando comparado a um método tradicional como a criação de histogramas. Além de não ser necessário realizar a extração de características de forma manual, ou

usando um algoritmo extrator em paralelo, a rede neural convolucional realiza a extração das características automaticamente, como parte do processo de aprendizagem profunda.

O aplicativo desenvolvido para validar o método de classificação é de fácil manuseio, intuitivo e não necessita de recursos de hardware avançados para ser executado. Dessa forma, diversos dispositivos podem fazer uso do aplicativo, mesmo aqueles que não possuem hardware avançado. Além disso, outros softwares podem usar o serviço fornecido pelo método computacional proposto, visto que é necessário apenas conexão via protocolo HTTP com o servidor, e a imagem deve ser convertida no padrão Base64, e depois enviada para o servidor, que irá responder a solicitação com as informações da praga, caso consiga encontra-la na imagem. Ao publicar o serviço de classificação de pragas dessa maneira, é simples fazermos com que demais softwares utilizem esse serviço, podemos por exemplo, acoplar essa tecnologia em drones, que capturam imagens e essas podem ser classificadas por meio do método proposto. Em suma, qualquer dispositivo capaz de se conectar a outro via protocolo HTTP pode utilizar o serviço de classificação de pragas, independentemente de seu poder de processamento, já que o processamento propriamente dito ocorre no lado do servidor. Isso facilita e muito o uso do serviço de classificação de pragas, o que faz dele uma alternativa de uso viável em aparelhos do tipo “internet das coisas”.

A classificação de pragas é um desafio pois muitas espécies são bastante parecidas, e sua classificação é difícil mesmo quando feita por especialistas. Contudo, a utilização de sistemas computacionais na resolução desse problema demonstrou-se confiável ao aplicar uma CNN especializada nesse tipo de problema.

Com relação aos trabalhos futuros, é interessante aumentar a base de imagens para ser usada no treinamento da CNN, seja com mais amostras das espécies já treinadas ou com novas espécies de pragas.

7 REFERÊNCIAS

- ABADI, Martín et al. Tensorflow: a system for large-scale machine learning. In: **OSDI**. 2016. p. 265-283.
- ARTZ, Alice F.; ARMOUR-THOMAS, Eleanor. Development of a cognitive-metacognitive framework for protocol analysis of mathematical problem solving in small groups. **Cognition and instruction**, v. 9, n. 2, p. 137-175, 1992.
- AGROLINK. **Pragas causam perdas de até R\$ 55 bilhões à agricultura no Brasil**. Agrolink, 2015. Disponível em: <https://www.agrolink.com.br/noticias/pragas-causam-perdas-de-ate-r-55-bilhoes-a-agricultura-no-brasil_220429.html/>. Acesso em: 05 ago. 2017.
- BRADSKI, G.; KAEHLE, A. **Learning OpenCV**. Sebastopol: O'Reilly. 2008.
- CHAPELLE, Olivier; SCHOLKOPF, Bernhard; ZIEN, Alexander. Semi-supervised learning (chappelle, o. et al., eds.; 2006)[book reviews]. **IEEE Transactions on Neural Networks**, v. 20, n. 3, p. 542-542, 2009.
- CHIMELO RUIZ, Luis Fernando; TEN CATEN, Alexandre; DINIZ DALMOLIN, Ricardo Simão. Árvore de decisão e a densidade mínima de amostras no mapeamento da cobertura da terra. **Ciência Rural**, v. 44, n. 6, 2014.
- CHOLLET, François. Xception: Deep learning with depthwise separable convolutions. **arXiv preprint**, p. 1610.02357, 2017.
- CIREŞAN, Dan C. et al. Mitosis detection in breast cancer histology images with deep neural networks. In: **International Conference on Medical Image Computing and Computer-assisted Intervention**. Springer, Berlin, Heidelberg, 2013. p. 411-418.
- COLLOBERT, Ronan; WESTON, Jason. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: **Proceedings of the 25th international conference on Machine learning**. ACM, 2008. p. 160-167.
- DE OLIVEIRA, Anderson Castro Soares et al. Aplicação de redes neurais artificiais na previsão da produção de álcool. 2010.
- DE SOUTO, M. C. P. et al. Técnicas de aprendizado de máquina para problemas de biologia molecular. **Sociedade Brasileira de Computação**, 2003.
- DILENA, Pietro; NAGATA, Ken; BALDI, Pierre. Deep architectures for protein contact map prediction. **Bioinformatics**, v. 28, n. 19, p. 2449-2457, 2012.

DOS SANTOS, Cicero; GATTI, Maira. Deep convolutional neural networks for sentiment analysis of short texts. In: **Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers**. 2014. p. 69-78.

ESTEVA, Andre et al. Dermatologist-level classification of skin cancer with deep neural networks. **Nature**, v. 542, n. 7639, p. 115, 2017.

FACON, Jaques. **Princípios Básicos da Visão por Computador e do Processamento de Imagens**. 2002.

FACON, J; **Processamento e Análise de Imagens**. Pontifícia Universidade Católica do Paraná. 2005.

GOLDSBOROUGH, Peter. A tour of tensorflow. **arXiv preprint arXiv:1610.01178**, 2016.

GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens**. São Paulo:Pearson Prentice Hall. 2010.

GÜZEL, Mehmet Serdar. A Hybrid Feature Extractor using Fast Hessian Detector and SIFT. **Technologies**, v. 3, n. 2, p. 103-110, 2015.

HASSANNEJAD, Hamid et al. Food image recognition using very deep convolutional networks. In: **Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management**. ACM, 2016. p. 41-49.

HIJAZI, Samer; KUMAR, Rishi; ROWEN, Chris. Using convolutional neural networks for image recognition. **Cadence Design Systems Inc.: San Jose, CA, USA**, 2015.

HINTON, Geoffrey et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. **IEEE Signal processing magazine**, v. 29, n. 6, p. 82-97, 2012.

HOWARD, Andrew G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.

IRUM, Isma et al. A nonlinear hybrid filter for salt & pepper noise removal from color images. **Journal of applied research and technology**, v. 13, n. 1, p. 79-85, 2015.

JIA, Yangqing et al. Caffe: Convolutional architecture for fast feature embedding. In: **Proceedings of the 22nd ACM international conference on Multimedia**. ACM, 2014. p. 675-678.

KARAMI, Ebrahim; SHEHATA, Mohamed; SMITH, Andrew. Image Identification Using SIFT Algorithm: Performance Analysis against Different Image Deformations. **arXiv preprint arXiv:1710.02728**, 2017.

KOHAVI, Ron. Glossary of terms. **Special issue on applications of machine learning and the knowledge discovery process**, v. 30, n. 271, p. 127-132, 1998.

KRATHWOHL, David R. A revision of Bloom's taxonomy: An overview. **Theory into practice**, v. 41, n. 4, p. 212-218, 2002.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. 2012. p. 1097-1105.

LAWRENCE, Steve et al. Face recognition: A convolutional neural-network approach. **IEEE transactions on neural networks**, v. 8, n. 1, p. 98-113, 1997.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, v. 521, n. 7553, p. 436, 2015.

LEONARDO, Matheus Macedo et al. Mid-level Image Representation for Fruit Fly Identification (Diptera: Tephritidae). In: **e-Science (e-Science), 2017 IEEE 13th International Conference on**. IEEE, 2017. p. 202-209.

LIU, Yun et al. Detecting cancer metastases on gigapixel pathology images. **arXiv preprint arXiv:1703.02442**, 2017.

LITER, Jeffrey C.; BÜLTHOFF, Heinrich H. An introduction to object recognition. **Zeitschrift für Naturforschung C**, v. 53, n. 7-8, p. 610-621, 1998.

LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2015. p. 3431-3440.

LORENA, Ana Carolina; DE CARVALHO, André CPLF. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.

LOWE, David G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, v. 60, n. 2, p. 91-110, 2004.

MARENGONI, Maurício; STRINGHINI, Stringhini. Tutorial: Introdução à visão computacional usando opencv. **Revista de Informática Teórica e Aplicada**, v. 16, n. 1, p. 125-160, 2009.

- MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport. 1999.
- MCCOOL, Chris; PEREZ, Tristan; UPCROFT, Ben. Mixtures of lightweight deep convolutional neural networks: applied to agricultural robotics. **IEEE Robotics and Automation Letters**, v. 2, n. 3, p. 1344-1351, 2017.
- MOGHADDAM, Baback; NASTAR, Chahab; PENTLAND, Alex. A Bayesian similarity measure for deformable image matching. **Image and Vision computing**, v. 19, n. 5, p. 235-244, 2001.
- O “alarmante” uso de agrotóxicos no Brasil atinge 70% dos alimentos. El País. Disponível em: <https://brasil.elpais.com/brasil/2015/04/29/politica/1430321822_851653.html>. Acesso em: 10 nov. 2018.
- PERELMUTER, Guy et al. Reconhecimento de imagens bidimensionais utilizando Redes Neurais Artificiais. **Anais do VIII SIBGRAPI**, p. 197-203, 1995.
- RAMCHARAN, Amanda et al. Deep learning for image-based cassava disease detection. **Frontiers in plant science**, v. 8, p. 1852, 2017.
- RAMPASEK, Ladislav; GOLDENBERG, Anna. Tensorflow: Biology’s gateway to deep learning?. **Cell systems**, v. 2, n. 1, p. 12-14, 2016.
- RUSSELL, Kimberly N. et al. Introducing SPIDA-web: wavelets, neural networks and Internet accessibility in an image-based automated identification system. **Automated taxon identification in systematics: theory, approaches and applications**, p. 131-152, 2007.
- SOLEM, Jan Erik. **Programming Computer Vision with Python: Tools and algorithms for analyzing images**. " O'Reilly Media, Inc.", 2012.
- SIMPSON, George Gaylord. Principles of animal taxonomy. 1961.
- SIVIC, Josef; ZISSERMAN, Andrew. Video Google: A text retrieval approach to object matching in videos. In: null. IEEE, 2003. p. 1470.
- SNEATH, Peter HA et al. **Numerical taxonomy. The principles and practice of numerical classification**. 1973.
- SRIVASTAVA, Nitish et al. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, v. 15, n. 1, p. 1929-1958, 2014.

STEINHAGE, V. et al. ABIS: automated identification of bee species. In: **BIOLOG Workshop, German Programme on Biodiversity and Global Change Status Report, Bonn**. 2001. p. 194-195.

SZEGEDY, Christian et al. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2015. p. 1-9.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. **Introdução ao datamining: mineração de dados**. Ciência Moderna, 2009.

VAN DER WALT, Stefan et al. scikit-image: image processing in Python. **PeerJ**, v. 2, p. e453, 2014.

WANG, Guan; SUN, Yu; WANG, Jianxin. Automatic image-based plant disease severity estimation using deep learning. **Computational intelligence and neuroscience**, v. 2017, 2017.

WALLACH, Izhar; DZAMBA, Michael; HEIFETS, Abraham. AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. **arXiv preprint arXiv:1510.02855**, 2015.

WATSON, A. T.; O'NEILL, M. A.; KITCHING, I. J. A qualitative study investigating automated identification of living macrolepidoptera using the Digital Automated Identification SYstem (DAISY). **Systematics and Biodiversity**, v. 1, n. 1, p. 287-300, 2003.

WICHMANN, Felix A.; SHARPE, Lindsay T.; GEGENFURTNER, Karl R. The contributions of color to recognition memory for natural scenes. **Journal of Experimental Psychology: Learning, Memory, and Cognition**, v. 28, n. 3, p. 509, 2002.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining – Practical Machine Learning Tools and Techniques – Third Edition**. Burlington: Morgan Kaufmann. 2011.

XIA, Xiaoling; XU, Cui; NAN, Bing. Inception-v3 for flower classification. In: **Image, Vision and Computing (ICIVC), 2017 2nd International Conference on**. IEEE, 2017. p. 783-787.

ZAYAS, I. Y.; FLINN, PAUL W. Detection Of Insects In Bulkwheat Samples With Machine Vision. **Transactions of the ASAE**, v. 41, n. 3, p. 883, 1998.

ZANNI, Luca; SERAFINI, Thomas; ZANGHIRATI, Gaetano. Parallel software for training large scale support vector machines on multiprocessor systems. **Journal of Machine Learning Research**, v. 7, n. Jul, p. 1467-1492, 2006.

ZEILER, Matthew D.; FERGUS, Rob. Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer, Cham, 2014. p. 818-833.