

UNIVERSIDADE ESTADUAL DO RIO GRANDE DO SUL

ENGENHARIA DE COMPUTAÇÃO

LORENZO LEOPOLDINO ANSOLIN

**SISTEMA COMPUTACIONAL PARA DETECÇÃO AUTOMÁTICA DE
DOENÇAS NA PRODUÇÃO DE UVA UTILIZANDO REDES NEURAIAS
CONVOLUCIONAIS**

GUAÍBA

2019

LORENZO LEOPOLDINO ANSOLIN

**SISTEMA COMPUTACIONAL PARA DETECÇÃO AUTOMÁTICA DE
DOENÇAS NA PRODUÇÃO DE UVA UTILIZANDO REDES NEURAS
CONVOLUCIONAIS**

Trabalho de Conclusão de Curso.
Apresentado como requisito parcial para
obtenção do título de Bacharel em
Engenharia de Computação na
Universidade Estadual do Rio Grande do
Sul.

Orientadora: Profa. Dra. Letícia Vieira
Guimarães

GUAÍBA

2019

LORENZO LEOPOLDINO ANSOLIN

**SISTEMA COMPUTACIONAL PARA DETECÇÃO AUTOMÁTICA DE
DOENÇAS NA PRODUÇÃO DE UVA UTILIZANDO REDES NEURAIS
CONVOLUCIONAIS**

Trabalho de Conclusão de Curso.
Apresentado como requisito parcial para
obtenção do título de Bacharel em
Engenharia de Computação na
Universidade Estadual do Rio Grande do
Sul.

Aprovada em 13/12/2019

BANCA EXAMINADORA:

Profa. Dra. Adriane Parraga
Universidade Estadual do Rio Grande do Sul

Prof. Dr. Celso Maciel da Costa
Universidade Estadual do Rio Grande do Sul

Profa. Dra. Leticia Vieira Guimarães
Universidade Estadual do Rio Grande do Sul



Lei nº. 11.646, de 10 de julho de 2001.

Parecer CEED nº. 1.150/2002

Reitoria - Rua Sete de Setembro, 1156.

90010-030 - Porto Alegre (RS)

www.uergs.edu.br

Reitora: Prof. Dr. Leonardo Alvim Beroldt da Silva

Coordenação de Curso: Profa. Dra. Letícia Vieira Guimarães

ATA DE AVALIAÇÃO

Aos treze dias do mês de dezembro de dois mil e dezenove realizou-se na Unidade de Guaíba a avaliação do Trabalho de Conclusão de Curso II do Curso de Engenharia de Computação, sob orientação do Professora Doutora Letícia Vieira Guimarães, do aluno:

LORENZO LEOPOLDINO ANSOLIN,

Intitulado:

"SISTEMA COMPUTACIONAL PARA DETECÇÃO AUTOMÁTICA DE DOENÇAS NA PRODUÇÃO DE UVA UTILIZANDO REDES NEURAIAS CONVOLUCIONAIS".

A banca examinadora foi composta pelos Professores Adriane Parraga e Celso Maciel da Costa que conferiram o conceito (A), ao trabalho apresentado, que fica arquivado nesta Unidade.


Prof. Dra. Adriane Parraga


Prof. Dr. Celso Maciel da Costa

Guaíba, 13 de dezembro de 2019.

Este trabalho dedico a minha família por todo incentivo e ajuda para que isso fosse possível.

RESUMO

Este trabalho apresenta uma metodologia para classificação automática de doenças em uma cultura de videira. A identificação ocorre diferenciando as principais doenças na cultura da uva, para isso faz uso de características específicas, como por exemplo manchas presentes na folha. As folhas contêm indicadores eficientes e que fornecem os sintomas das doenças atingíveis. Este classificador desenvolvido fornece o diagnóstico da doença que aflige uma videira e reduzindo o tempo e os custos no processo. A identificação e diagnóstico de doenças de forma automática possibilita uma ação mais rápida no sentido de ser mais eficaz na eliminação da doença. O modelo desenvolvido para classificação das doenças utiliza redes neurais convolucionais, sendo a camada de entrada do sistema os dados de folhas com determinadas doenças, e na camada de saída a doença mais provável assim como a probabilidade de acerto do sistema. Como resultado, é desenvolvida a importância de cada parâmetro de uma rede neural convolucional e a importância no contexto geral, influenciando no desempenho final. A arquitetura MobileNet foi utilizada e parâmetros ajustados permitiram taxas de erros muito próximos de zero.

Palavras-chaves: Redes Neurais Convolucionais, Agricultura de precisão, MobileNet.

ABSTRACT

This work presents a methodology for automatic classification of diseases in a vine crop. Identification occurs by differentiating itself as the main diseases in grape cultivation, for that it makes use of specific characteristics, as examples of spots present in the leaf. The leaves selected indicative efficient and the symptoms of the diseases affectable. This developed classifier provides the diagnosis of disease that affects a vine and reduces the time and costs in the process. Automatically identifying and diagnosing disease enables faster, meaningless action to be more effective at removing disease. The model developed for disease classification uses convolutional neural networks, being a system input layer or disease leaf data, and a more likely disease output layer, as well as the probability of system execution. As a result, an important value of each neural convolutional network parameter is developed and a general context value influencing the final performance. The MobileNet architecture was used and tuned to allow error rates very close to zero.

Keywords: Convolutional Neural Networks, Precision Farming, MobileNet.

LISTA DE FIGURAS

Figura 1 –Exemplificação do banco de dados	10
Figura 2 –Diagrama de procedimentos	14
Figura 3 –Visão Geral.....	16
Figura 4 –Arquitetura VGG 16	19
Figura 5 – Camadas arquitetura VGG16	20
Figura 6 –Camadas <i>dephtwise</i>	21
Figura 7 – Exemplificação de uma convolução simples.....	24
Figura 8 – Exemplificação de uma convolução com <i>ZeroPadding</i>	24
Figura 9 – <i>Dropout</i> em uma rede neural	27
Figura 10 – Operação de <i>Dropout</i>	28
Figura 11 – Imagem gerada após pré-processamento	31
Figura 12 – Histograma de uma Imagem da base de dados.....	35
Figura 13 – Relação entre função perda e épocas.....	35
Figura 14 – Relação entre acurácia do modelo e número de épocas	35
Figura 15 – Matriz de confusão	36
Figura 16 – Visão Geral de uma GAN.....	40

LISTA DE TABELAS

Tabela 1 – Arquitetura da MobileNet.....	24
Tabela 2 – Divisão dos dados.....	32

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	TEMA E PROBLEMA.....	13
1.2	OBJETIVOS.....	14
1.2.1	Objetivo Geral.....	14
1.2.2	Objetivos Específicos.....	14
1.3	JUSTIFICATIVA.....	15
2	MÉTODOS DE IDENTIFICAÇÃO DE DOENÇAS.....	16
3	METODOLOGIA.....	17
3.1	BANCO DE DADOS E PRÉ-PROCESSAMENTO.....	17
3.2	REDE NEURAL CONVOLUCIONAL.....	19
3.3	IDENTIFICAÇÃO DOS RESULTADOS.....	19
4	REDES NEURAIAS CONVOLUCIONAIS (CNN).....	20
4.1	ARQUITETURA.....	20
4.1.1	VGG16.....	21
4.1.2	MobileNets.....	22
4.2	CAMADAS.....	25
4.2.1	Camada de Entrada (<i>Input Layer</i>)	25
4.2.2	<i>Zero Padding</i>	25
4.2.3	Convolução 2D.....	26
4.2.4	<i>Batch Normalization</i>	27
4.2.5	ReLu.....	28
4.2.6	<i>Pooling 2D</i>	28
4.2.7	<i>Dropout</i>	29

4.2.8	Classificação <i>SoftMax</i>	30
5	IMPLEMENTAÇÃO.....	31
5.1	BASE DE DADOS.....	31
5.2	PRÉ-PROCESSAMENTO.....	32
5.3	SEGMENTAÇÃO DO PRÉ-PROCESSAMENTO.....	33
5.4	PARÂMETROS PARA TREINAMENTO DA REDE NEURAL CONVOLUCIONAL.....	34
5.4.1	Otimizador.....	35
5.4.2	Taxa de Aprendizado.....	35
5.4.3	Métrica.....	37
5.4.4	Função de perda (<i>Loss Function</i>).....	37
6	RESULTADOS.....	38
6.1	AVALIAÇÃO DOS RESULTADOS UTILIZANDO MOBILENET.....	38
7	CONCLUSÃO.....	41
7.1	TRABALHOS FUTUROS.....	41
	REFERÊNCIAS.....	43

INTRODUÇÃO

A agricultura familiar, urbana e domiciliar são as que mais sofrem com as doenças que atingem suas culturas, isso ocorre porque os pequenos produtores estão envolvidos em um ambiente muito competitivo e não possuem tantos recursos quanto os grandes produtores. Quando alguma doença atinge sua plantação é necessário identificar a doença afetada, isto não é uma tarefa trivial, sendo necessário contratar especialistas para identificar o que foi afetado e sugerir uma possível solução de tratamento. Na maioria dos casos não são realizadas consultas com especialistas, isto pode ocasionar prejuízos e danos irreparáveis, além da aplicação de defensivos de maneira desmedida, aumentando o impacto ambiental. Em virtude do que foi mencionado, percebe-se que a agricultura necessita de constante inovação e ferramentas que simplifiquem a tarefa de pequenos produtores, a partir disso desenvolveu-se uma aplicação que através de uma fotografia contendo a folha atingida pela doença será possível identificá-la automaticamente utilizando algoritmos de aprendizado de máquina, Neste contexto, o projeto envolve a detecção automatizada das principais doenças na cultura da uva. Para especificar a doença que afeta determinada cultura é utilizado um grande banco de imagens que através de aprendizado de máquina e redes neurais, caracterizará as doenças, identificando na fotografia a referente doença.

Redes neurais são sistemas computacionais compostos por nós interconectados, formando um neurônio. Cada nó será responsável por extrair uma característica nos dados da camada de entrada e este nó irá prover para a camada seguinte uma nova extração de características baseada na anterior. Em uma rede neural, existem diversos parâmetros entre eles estão características das camadas,

função de custo, otimizador, entre outros. Compreender estes parâmetros e configurá-los faz parte do escopo deste trabalho.

1.1 TEMA E PROBLEMA

A identificação de doenças através de folhas é um processo de interpretação, sendo uma atividade complexa. O uso de técnicas de processamento de imagem e redes neurais se faz necessário para validar diagnósticos.

Na Figura 1, nota-se que existem diversas doenças que se manifestam de maneira muito semelhante, tornando a tarefa de diagnosticar em algo realmente complexo.

Figura 1: Exemplificação do banco de dados.



(a) Mancha das folhas (*Isariopsis Clavispora*).

(b) Podridão-Negra (*Guignardia bidwellii*)

(c) Esca (Diversas espécies de fungicidas)

Fonte: MOHANTY; HUGHES; SALATHÉ (2016).

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver uma ferramenta para identificar e classificar doenças de uma videira a partir da imagem de uma folha, utilizando redes neurais convolucionais.

1.2.2 Objetivos Específicos

- Analisar as principais camadas de uma rede neural convolucional.
- Condicionar o pré-processamento, envolvendo normalização dos dados, análise para validar imagens, entre outros.
- Utilizar algoritmo de aprendizado de máquina utilizando redes neurais convulsionais.

1.3 JUSTIFICATIVA

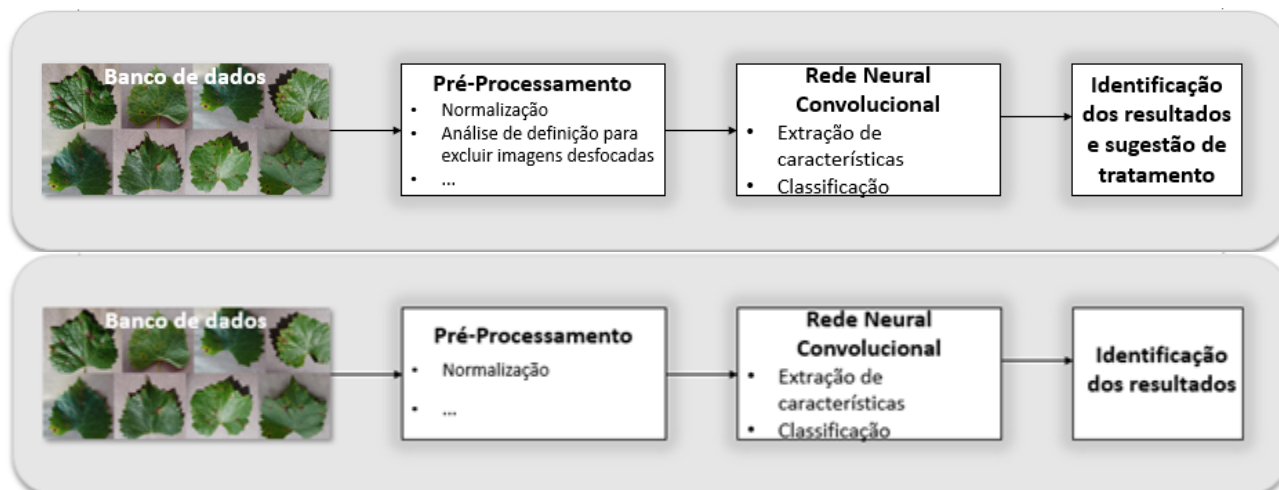
Engenharia de computação envolve buscar soluções para problemas em diversas áreas na sociedade aplicando tecnologia de computação – software e hardware. Assim, o diagnóstico de doenças da videira é um auxílio à segurança alimentar, pois a produção de alimentos por pequenos produtores corresponde à 80% da produção agrícola mundial (UNEP, 2013), além disso, estipula-se que mais de 50% das perdas da produção são ocasionadas por doenças e pestes (Harvey et al., 2014). O uso de uma ferramenta tecnológica e automatizada para diagnosticar estas doenças pode reduzir as perspectivas de perdas na produção.

2 MÉTODOS DE IDENTIFICAÇÃO DE DOENÇAS

As primeiras abordagens envolvem a detecção de doenças utilizando redes neurais convolucionais compreendem o trabalho de MOHANTY; HUGHES; SALATHÉ (2016), abordando a detecção automática das doenças, utilizando um banco de dados voltado para a agricultura alimentícia, destacando uma possível crise alimentar. Neste trabalho, a inteligência artificial desenvolvida utiliza arquitetura AlexNet (Krizhevsky et al., 2012) e GooGleNet (Szegedy et al., 2015). Ele relata também o avanço que a visão computacional vem apresentando, e seu potencial, como a utilização de smartphones para ajudar na identificação de doenças, dispendo de seu poder computacional e acessórios, como câmeras, etc. Utilizando as ferramentas citadas, o autor desenvolve uma rede neural convolucional fazendo uso de diversas configurações, alternando o modelo de testes e de treinamento, além do número de camadas desta rede neural. Os melhores resultados obtidos pelo autor após diversos testes foram de 99,27% de taxa de assertividade. Estes valores são muito satisfatórios e úteis pois indicam que é concebível alcançar grandes taxas de sucesso mesmo utilizando novas ferramentas computacionais, mais robustas, que permitem testar diversas configurações e arquiteturas, potencializando os resultados.

3 METODOLOGIA

Figura 2: Diagrama de procedimentos.



Fonte: Elaborada pelo autor.

3.1 BANCO DE DADOS E PRÉ-PROCESSAMENTO

Este bloco representa a base de dados em si e a aquisição de dados para serem alimentados na camada de entrada da rede neural convolucional. Em uma rede neural convolucional, a complexidade para extração de características é mitigada através de um banco de dados robusto. O banco de dados utilizado é Mohanty/PlantVillage, composto por imagens de folhas saudáveis e outras três categorias de doenças, somando-se todas as categorias, o banco de dados possui 4062 imagens com resolução de 256x256 pixels. Estas imagens estão distribuídas entre culturas e doenças diversas. As imagens do banco de dados já estão redimensionadas, a resolução de escolha foi definida como a menor resolução que fornece precisão satisfatória para segmentar a representação das doenças nas folhas. A utilização de resoluções maiores do que a mínima satisfatória, resultará

em uma rede neural que extrai características muito mais específicas e pouco úteis para esta aplicação.

A primeira etapa no desenvolvimento de uma rede neural envolve a aquisição de dados, estes dados foram separados em grupos de: teste, validação e treino, mantendo a proporção de aproximadamente 70% das imagens no grupo de treino, 15% no grupo de validação e 15% no grupo de testes. O grupo de treino compõe as imagens que serão alimentadas no treinamento da rede neural. O grupo de validação fornece uma análise imparcial do modelo treinado com o grupo de treino, ajustando os parâmetros do modelo. O grupo de testes pode ser considerada como uma outra camada do grupo de treino.

As imagens dentro dos respectivos conjuntos são colocadas em ordem aleatória, evitando possíveis tendências relacionadas à ordem das imagens.

Para o desenvolvimento do trabalho, foi utilizada a linguagem de programação Python. As bibliotecas empregadas foram:

- Numpy: Utilizada principalmente para realizar cálculos em vetores Multidimensionais.
- OpenCV: Utilizada para desenvolvimento na área de visão computacional.
- Keras: Biblioteca para prototipagem de redes neurais.
- Sklearn/scikit: Módulo utilizado para aprendizado de máquina.
- Matplotlib: Biblioteca útil para incorporar gráficos e realizar análises.

3.2 REDE NEURAL CONVOLUCIONAL

A etapa da rede neural convolucional pode ser compreendida como a exemplificação de como será construída e suas principais etapas.

A configuração da rede neural pode ser dividida como: Determinação da topologia a ser utilizada e determinação dos parâmetros para o treinamento. Antes do treinamento é necessário definir a Função de custo, que é está relacionada com o quão rápido os pesos dos neurônios vão convergir para o valor esperado. Existem diversas funções de custo, as mais utilizadas são: *cross-entropy loss* e erro quadrático médio. Outro parâmetro são os otimizadores que aprimoram a atualização dos pesos, utilizando por exemplo a taxa de aprendizado.

3.3 IDENTIFICAÇÃO DOS RESULTADOS

A etapa de avaliar resultados pode ser exemplificada como a maneira pela qual será testada a rede neural, e como será validada. Durante esta fase, o conjunto de testes é utilizado e determinará a confiabilidade da rede com os dados dos grupos de validação e testes.

4 REDES NEURAIAS CONVOLUCIONAIS (CNN)

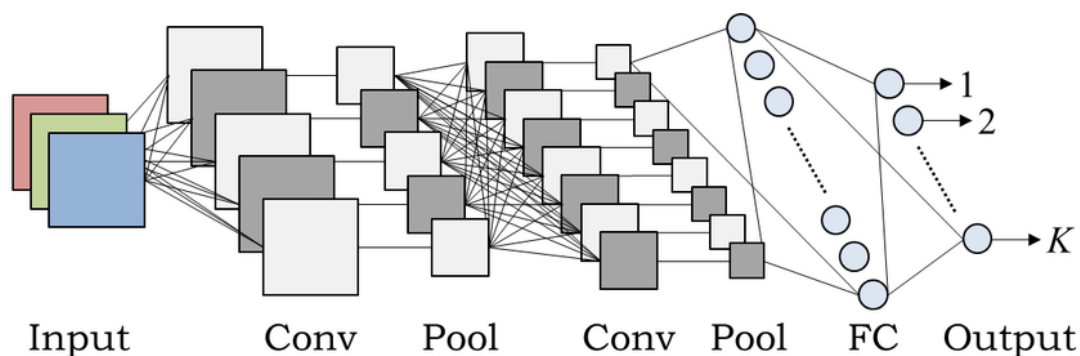
A identificação de parâmetros importantes na caracterização de um objeto ou região em uma imagem para classificação pode ser uma tarefa muito custosa e extremamente complexa. Uma das maiores dificuldades da visão computacional é generalizar as variações reais da imagem. As redes neurais convolucionais no contexto de inteligência artificial e aprendizado de máquina, fornecem a capacidade de extrair características que muitas vezes não seriam observadas se comparado à atividade humana. Neste capítulo serão abordados os principais componentes de uma CNN, aspectos de arquitetura e características específicas.

4.1 ARQUITETURA

No âmbito de redes neurais convolucionais, a presença de arquiteturas (modelos) para aprendizado de máquina tornou-se populares desde o primeiro modelo proposto (AlexNet), desde então foram desenvolvidos novos modelos, mais complexos buscando diminuir o erro. Porém além da exatidão, surgiram novos requisitos, estes dependem da aplicação, como por exemplo limitações de espaço físico, embarcados em drones. Estes novos requisitos requerem uma alta exatidão, porém também um tempo de resposta satisfatório, mesmo que em hardwares com pouco poder de processamento.

Atualmente existe uma infinidade de arquiteturas diferentes, porém sua maioria segue um mesmo modelo base, que pode ser observado na Figura 4.

Figura 4: Arquitetura VGG16.



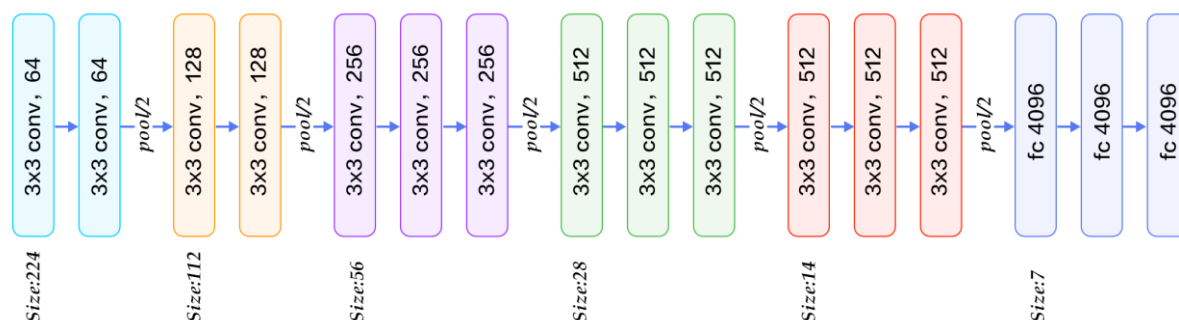
Fonte: Simonyan, Karen (2015)

Esta é uma representação de uma das arquiteturas de CNN mais utilizadas, nela pode-se notar que a CNN é subdividida em camadas, sendo elas, camada de entrada, responsável por alimentar o sistema com as imagens de entrada, seguida por camadas profundas de Convolução, *Pooling*, totalmente conectada (FC) e por fim, a camada de classificação, que é a saída do sistema.

4.1.1 VGG16

A arquitetura VGG16 é característica por ser uma das primeiras arquiteturas a utilizar camadas de filtros menores do que os convencionais. A Arquitetura pode ser visualizada através da Figura 5. O uso de filtros 3x3, ao invés de filtros 7x7 presentes na AlexNet, permite um menor número de parâmetros. Com um menor número de parâmetros para aprender, melhor será para aumentar a velocidade com que ocorre a convergência, reduzindo também a chance do sistema em aprender parâmetros pouco úteis. (Simonyan, Karen, 2012).

Figura 5: Camadas da arquitetura VGG16.



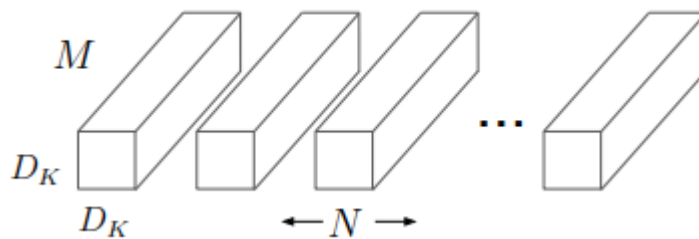
Fonte: Simonyan, Karen (2015)

4.1.2 MobileNets

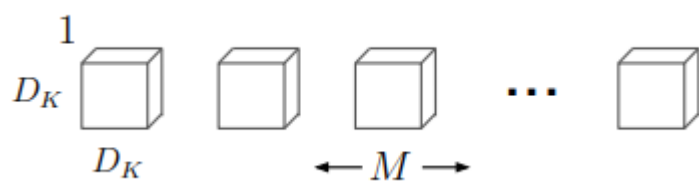
MobileNet é uma arquitetura para redes neurais convolucionais. Desenvolvida para maximizar a precisão, tendo em mente a restrição computacional, que pode estar presente, por exemplo, em dispositivos embarcados. Esta arquitetura é baseada em um grupo de dois parâmetros que podem ser redimensionados para suprir as necessidades da aplicação.

As primeiras camadas desta arquitetura utilizam a técnica baseada em *depthwise separable convolutions*, separando os canais para convolução e segundo Howard, Andrew (2017), Isto reduz o custo computacional das primeiras camadas. É aplicado um filtro 1x1 para combinar as saídas da camada de convolução *depthwise* (dw), este filtro combina as entradas em um novo grupo de saídas. A camada de *depthwise separable convolutions* divide cada camada em duas, uma para filtragem e a outra para combinar os dados. A figura 6 abaixo demonstra como a camada padrão de convolução (a) é fatorada em uma camada de *depthwise convolution* (b) e em uma camada 1x1 (c).

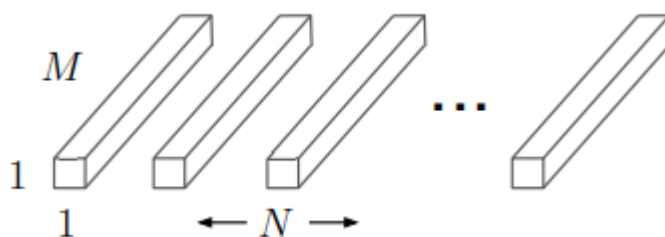
Figura 6: Camadas *depthwise*.



(a) Camada de convolução padrão.



(b) Camada de convolução *Depthwise*



(c) Camada de convolução 1x1, conhecida também por *pointwise convolution*

Fonte: Howard, Andrew (2017).

Na Tabela 1 abaixo, pode-se perceber nas suas respectivas colunas: As camadas desta arquitetura, seus respectivos kernels de saída.

Tabela 1: Arquitetura da MobileNet.

Layer (type)	Output Shape
input_2 (InputLayer)	(None, 224, 224, 3)
conv1_pad (ZeroPadding2D)	(None, 225, 225, 3)
conv1 (Conv2D)	(None, 112, 112, 32)
conv1_bn (BatchNormalization)	(None, 112, 112, 32)
conv1_relu (ReLU)	(None, 112, 112, 32)
conv_dw_1 (DepthwiseConv2D)	(None, 112, 112, 32)
● ● ●	
global_average_pooling2d_2 ((None, 1024)
reshape_1 (Reshape)	(None, 1, 1, 1024)
dropout (Dropout)	(None, 1, 1, 1024)
conv_preds (Conv2D)	(None, 1, 1, 1000)
reshape_2 (Reshape)	(None, 1000)
act_softmax (Activation)	(None, 1000)

Fonte: Elaborada pelo autor.

4.2 CAMADAS

No segmento abaixo, serão explicadas todas as camadas da arquitetura MobileNet e suas principais características.

4.2.1 Camada de entrada (Input Layer)

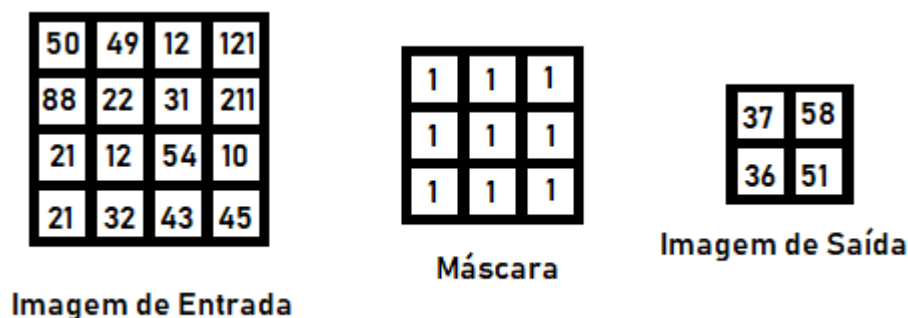
Uma rede neural convolucional tem como entrada uma matriz representando a imagem, as dimensões e a quantidade de canais contidas nesta matriz. Além disso o número de amostras também pode ser fornecido, este valor é conhecido como *batch size*. O *batch size* pode conter o valor total de amostras, porém ao utilizar um número elevado de imagens para treinamento, será necessário armazená-las em memória, o que pode ser inviável. É necessário subdividir o número de amostras e aumentar a quantidade de iterações, visando contemplar os dados em sua totalidade.

4.2.2 *ZeroPadding*

Quando se aplica convolução com uma determinada máscara sobre uma imagem, gera-se uma nova imagem, esta imagem tem sua dimensão reduzida. Para manter a resolução original utiliza-se a técnica *ZeroPadding*, ela adiciona zeros nas bordas da imagem. Para exemplificar este comportamento pode-se notar a Figura 7, nela percebe-se o comportamento de uma convolução simples com uma máscara de 3x3 e a saída da imagem com sua resolução reduzida para 2x2. Na Figura 8

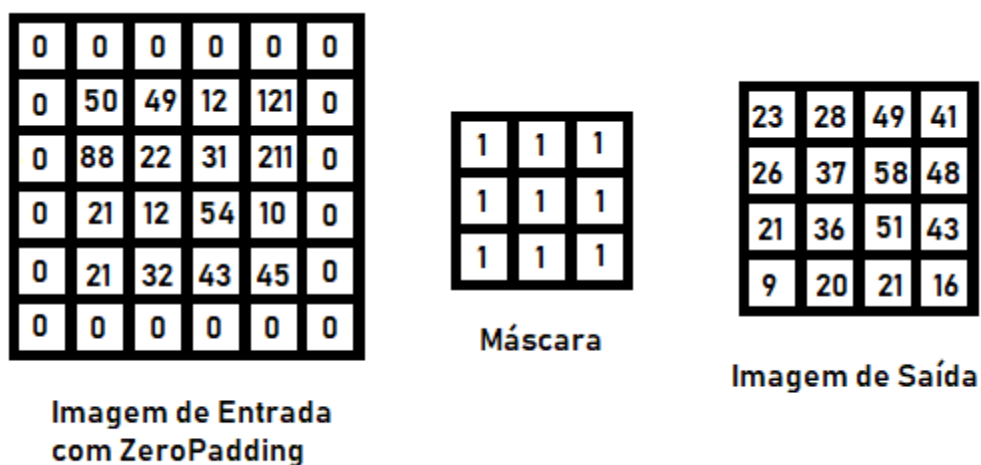
percebe-se o comportamento da convolução desta mesma máscara, porém aplicando *ZeroPadding*.

Figura 7: Exemplificação de uma convolução simples.



Fonte: Elaborada pelo autor.

Figura 8: Exemplificação de uma convolução com *ZeroPadding*.



Fonte: Elaborada pelo autor.

4.2.3 Convolução 2D

Um filtro linear espacial calcula a média aritmética ponderada local dos pixels da janela. Os pesos são definidos através de sub imagens denominadas de filtro,

máscara, núcleo, padrão, peso ou janela. (Gonzalez e Woods, 2002). Considerando a imagem f de tamanho $M \times N$, uma máscara w de tamanho $m \times n$. A filtragem linear pode ser descrita como:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (1)$$

No âmbito de redes neurais convolucionais, suas convoluções são indispensáveis, a partir delas é possível extrair o mapa de características. Durante o processo de treinamento, os pesos destes filtros são ajustados em cada época, através do conjunto de imagens contidas no *batch size*. Eles são calculados para que sejam ativados na presença de características relevantes, como orientação de bordas ou mancha de cores (KARPATHY, 2017).

4.2.4 *Batch Normalization*

Batch Normalization é uma camada em que ocorre a normalização dos dados. A preparação dos dados para treinamento pode ser feita de diversas maneiras, a mais simples envolve um típico processo de normalização que reescala os dados em um novo conjunto de amostragem, este processo pode ser problemático, uma vez que podem ocorrer diferenças gigantescas nas grandezas, fazendo com que conjuntos com valores muito altos possam encobrir conjuntos menores. O treinamento de uma rede neural pode ser muito custoso computacionalmente, normalizar as camadas de ativação reduz a complexidade, considerando isso, *batch normalization* utiliza a distribuição do somatório das

entradas de uma camada de ativação do tamanho de batch e calcula a média e variância, que são utilizadas para normalizar a camada de ativação.

4.2.5 ReLu

Existem diversas funções de ativação, porém todas elas possuem o mesmo objetivo, determinar a classificação da saída de determinado neurônio e conectar na saída de uma camada de ativação. Segundo Hinton e Krizhevskye Sutskever (2012), essa função determina se cada camada vai ter valor zero ou diferente de zero, a mesma forma que um neurônio transmite ou não pulsos elétricos.

Uma das funções de ativação mais utilizadas, e que foi utilizada é a ReLu (*Rectified Linear Units*), esta função de ativação é dada pela equação:

$$f(x) = \max(0, x) \quad (2)$$

A função de ativação ReLu se comporta de maneira com que quanto maior o valor de entrada, maior será a saída desta função de ativação.

4.2.6 Pooling 2D

As camadas de *pooling* tem como objetivo principal reduzir a dimensionalidade da rede, reduzindo assim o custo computacional, porém é necessário distinguir o que não é relevante e que pode ser suprimido ou reduzido, para isso, a técnica consiste em calcular o valor máximo de cada canal (região).

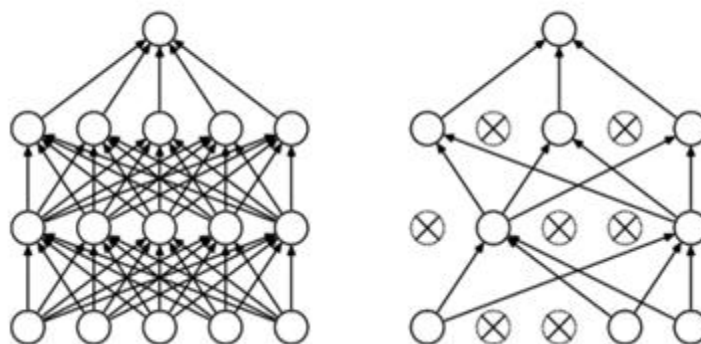
Desta forma, ele elimina valores desprezíveis, criando uma invariância a pequenas mudanças e distorções locais (Araújo, Carneiro, Silva, 2017).

4.2.7 Dropout

Operações de *Dropout* são úteis para evitar que o modelo extraia características muito específicas dos dados de teste, consequentemente perdendo a capacidade de generalizar em novos dados. A ideia principal desta técnica é o descarte aleatório de uma porcentagem de neurônios e os adicionando novamente na iteração seguinte. Essa técnica também confere à rede a habilidade de aprender atributos mais robustos, uma vez que um neurônio não pode depender da presença específica de outros neurônios (Araújo, Carneiro, Silva, 2017).

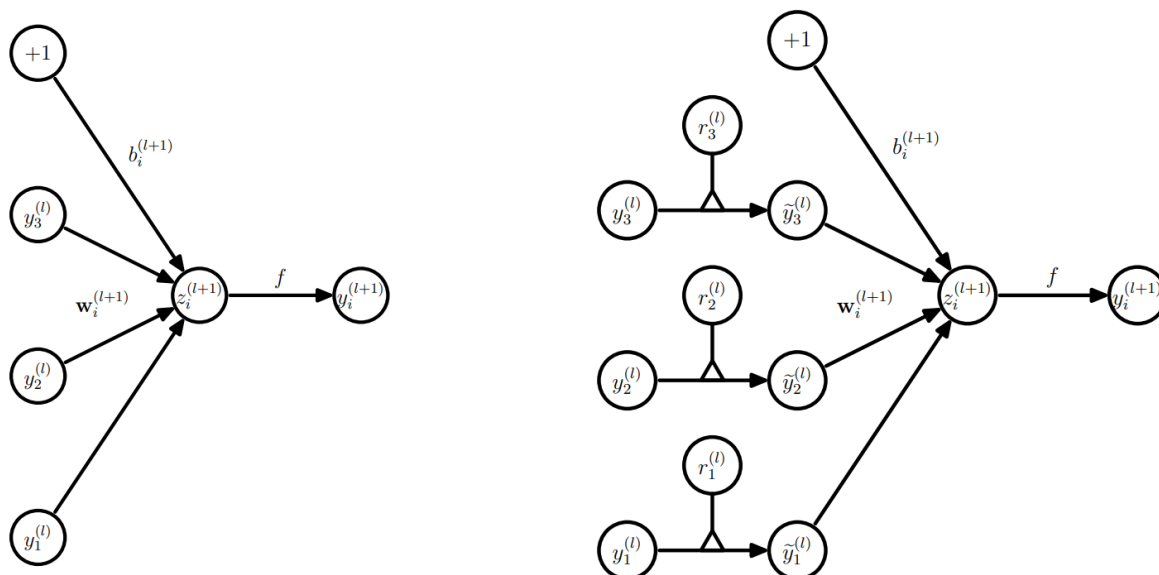
A figura 9 abaixo demonstra uma arquitetura genérica e a ação de exclusão de neurônios. Na figura 10 observa-se a comparação da operação convencional e com o uso da adição da camada de *Dropout*.

Figura 9: *Dropout* em uma rede neural.



Fonte: Srivastava, Nitish (2014).

Figura 10: Operação de *Dropout*.



Fonte: Srivastava, Nitish (2014).

4.2.8 Classificação *SoftMax*

A função de ativação *SoftMax* é comumente utilizada na camada de saída, servindo como uma camada de classificação. Esta camada calcula a distribuição de ocorrência das diversas classes.

5 IMPLEMENTAÇÃO

Este capítulo tem como objetivo detalhar o sistema para verificar a presença de doenças em uma videira. Nele será demonstrado as etapas do desenvolvimento e parâmetros de ajuste.

5.1 BASE DE DADOS

A base de dados de uma rede neural convolucional tem a função de fornecer as características que serão extraídas pela rede neural convolucional. Esta tarefa demanda uma quantidade considerável de imagens. A quantidade de imagens necessárias como entrada depende basicamente de dois fatores: 1) Número de características necessárias para a aprendizagem da rede e; 2) Quantas classes devem ser reconhecidas pelo sistema. A base de dados utilizada é a PlantVillage, possuindo 422 imagens de folhas de videiras saudáveis e 1075 com a doença mancha-das-folhas. A resolução das imagens é um dos responsáveis por definir quantas camadas e conseqüentemente a quantidade de parâmetros a serem aprendidos. Uma resolução muito alta poderá fazer com que a rede neural extraia características não muito relevantes, além de aumentar o tempo de processamento drasticamente. Para escolha da resolução das imagens, o princípio utilizado foi de manter a menor resolução possível, que contenha a maior parte das características das doenças. Para isso, a resolução de 226x226 pixels foi adotada. Este trabalho está muito associado à necessidade de ampliar o banco de dados, uma vez que existem mais de 10 doenças de uma videira que se manifestam através das folhas. Cabe acrescentar que ampliar a base de dados e adquirir volume de imagens pode

ser uma tarefa extremamente difícil, ainda mais considerando a cultura da uva, em que a colheita ocorre (na maioria das espécies) uma vez ao ano e as doenças afetam drasticamente e rapidamente a qualidade da colheita, sendo necessário agir rápido após diagnosticar a doença.

Diante deste cenário, se vê necessidade de métodos alternativos para ampliar a base de dados, ainda mais em casos de doenças com pouca ocorrência.

5.2 PRÉ PROCESSAMENTO

O pré-processamento envolve normalizar os dados para serem inseridos na camada de entrada. Nesta etapa está envolvido a utilização da técnica de *downsampling* que reduz a resolução das imagens para 224x224 pixels. Outro aspecto envolve segmentar a zona de interesse (folha), excluindo o fundo das imagens, esta tarefa garante ao sistema a capacidade de reduzir a quantidade de dados para serem treinados, conseqüentemente reduzindo a carga computacional e o tempo de processamento, já que a rede neural convolucional não terá de abstrair o objeto de interesse.

Figura 11: Imagem gerada após pré-processamento.



(a) Antes do pré-processamento.

(b) Após o pré-processamento

Fonte: Elaborada pelo autor.

5.3 SEGMENTAÇÃO DO PRÉ PROCESSAMENTO

Para segmentar as folhas utilizou-se um cálculo entre a relação das bandas, esta relação é um índice de vegetação, e é útil na análise de vegetação natural (presentes em zonas de preservação ambiental) e vegetação agrícola. Os índices de vegetação são responsáveis por realçar características como: vigor da planta, corpos de água, solo exposto, diferentes níveis de vegetação, entre outros.

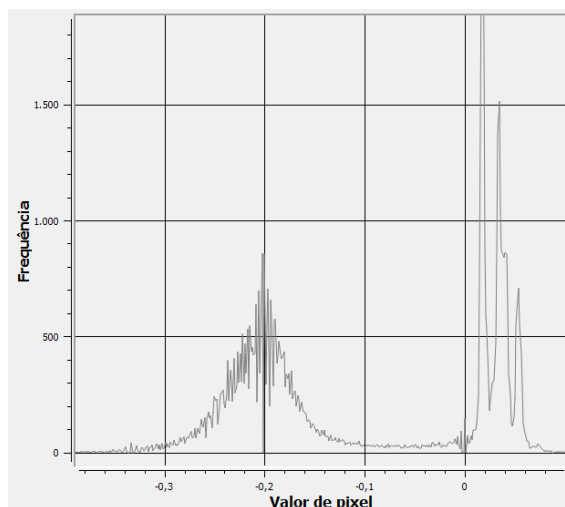
O índice de diferença normalizada Vermelho/Verde (*Redness Index*) admite valores de -1 à 1 e é dado pela equação:

$$Redness\ Index = \frac{R-G}{R+G} \quad (3)$$

Onde R é o valor de vermelho e G o valor do verde do pixel.

A escolha do limiar para segmentar o objeto de interesse, se deu através da análise de histograma de diversas imagens. O histograma de uma das imagens pode ser observado na Figura 12.

Figura 12: Histograma de uma Imagem da base de dados.



Fonte: Elaborada pelo autor.

Segundo Motohka, Takeshi (2010), vegetação, solo, água/neve possuem valores respectivamente positivos, negativos e próximos de zero, desta forma podemos admitir que um limiar igual à zero pode ser efetivo para distinguir vegetação e outros tipos de cobertura de solo.

5.4 PARÂMETROS PARA TREINAMENTO DA REDE NEURAL CONVOLUCIONAL

Para realizar o treinamento da rede neural convolucional, a base de dados foi dividida em dados de treino, validação e teste. A proporção pode ser observada através da Tabela 2 abaixo.

Tabela 2: Divisão dos dados.

Número	Classe	Treino (%)	Treino (Val)	Validação (%)	Validação (Val)	Teste (%)	Teste (Val)	Total (Val)
1	<i>Isariopsis</i>	88	950	6	63	6	63	1074
2	Saudável	70	297	15	63	15	63	421
#	Geral	158	1247	21	124	21	124	1495

Fonte: Elaborada pelo autor.

Nota-se que os dados referentes ao treinamento da classe *Isariopsis*, alocam uma maior quantidade de imagens, isso é justificável já que a morfologia das lesões tem maior diversidade e complexidade se apresentando de forma diferente nas várias imagens da classe *Isariopsis*., conseqüentemente maior quantidade de parâmetros devem ser aprendidos.

5.4.1 Otimizador

Otimizadores são algoritmos que podem ser utilizados para reduzir a função custo, localizando valores otimizados para os pesos de cada neurônio. O otimizador utilizado foi o Adam (Estimativa do momento adaptativo), que segundo Kingma, Diederik (2015), é um algoritmo para gradiente de primeira ordem baseado em otimizações estocásticas. O método computa individualmente a taxa de aprendizado adaptativo para diferentes parâmetros de estimativas do gradiente.

5.4.2 Taxa de aprendizado

A taxa de aprendizado se refere à taxa com que os pesos precisam ser atualizados durante o treinamento, sendo relacionada à velocidade com que o modelo aprende. Geralmente, um valor muito alto atribuído à esta taxa faz com que o modelo aprenda muito rápido, porém não garantirá que alcance os melhores valores de ajuste dos pesos de cada neurônio. Por outro lado, uma taxa com valor muito baixo permite com que o modelo aprenda mais, porém a um custo computacional maior. A taxa de aprendizado foi escolhida através de testes empíricos, com o valor de $\eta = 0,0001$.

5.4.3 Métrica

A métrica é uma função usada para julgar a performance do modelo. Em uma rede neural convolucional, a métrica utilizada é a *accuracy*, que relaciona os casos verdadeiros com o número total de casos, retornando um valor no intervalo de zero à um. Sua equação pode ser observada abaixo.

$$Exatidão = \frac{\text{casos corretos}}{\text{total de casos}} \quad (4)$$

5.4.4 Função de Perda (*loss function*)

Enquanto a métrica julga a performance do modelo, a função de perda é o meio pelo qual o sistema aprenderá. As principais funções de perda para classificação são *Cross Entropy* e *mean squared error*. Sua equação pode ser vista abaixo, considerando \hat{Y} como o valor predito e Y sendo o valor esperado. Suas equações são respectivamente:

$$L(Y, \hat{Y}) = - \sum_{i=0}^N \sum_{j=0}^M (Y_{ij} * \log(\hat{Y}_{ij})) \quad (5)$$

$$MSE(Y, \hat{Y}) = \frac{1}{n} \sum_{i=0}^n (Y_{ij} - (\hat{Y}_{ij}))^2 \quad (6)$$

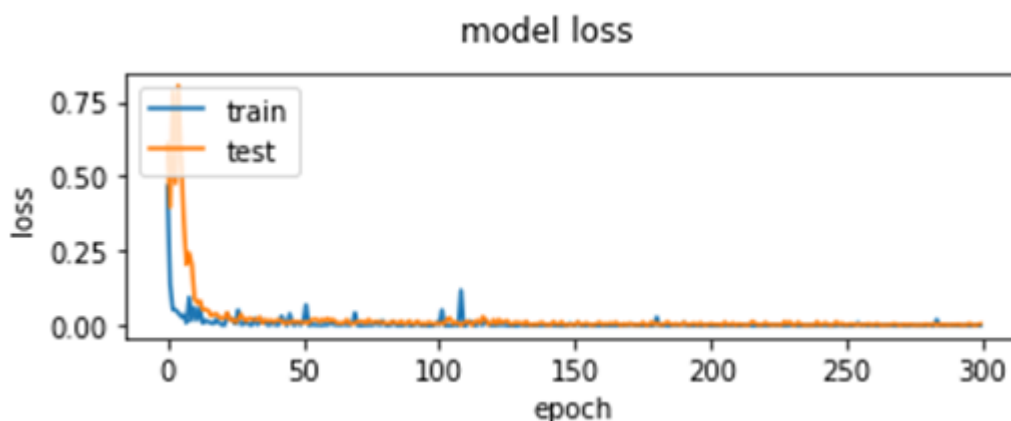
6 RESULTADOS

Neste capítulo serão apresentados os resultados obtidos com a rede neural convolucional através da arquitetura MobileNet.

6.1 AVALIAÇÃO DO RESULTADO UTILIZANDO MOBILENET

Na Figura 12 pode-se observar a relação entre a função perda e o número de épocas. A função perda utilizada é a *categorical crossentropy*, ela é muito útil em problemas de classificação onde apenas um resultado pode ser correto, que é o nosso caso, já que uma folha não pode simultaneamente apresentar mancha-das-folhas e ser saudável. *Categorical crossentropy* compara a distribuição das predições com a distribuição verdadeira. Sendo probabilidades próximas a um indicarem valores referentes ao verdadeiro, e próximo a zero o oposto.

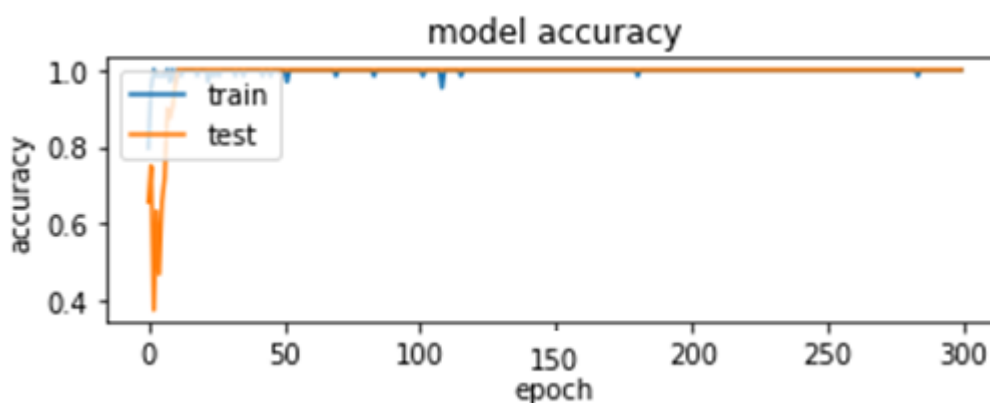
Figura 13: Relação entre função perda e épocas.



Fonte: Elaborada pelo autor.

Na Figura 13 está demonstrada a relação entre a acurácia do modelo e o número de épocas. É interessante perceber que a partir da época cinquenta, este conjunto de testes já está estabilizado, garantindo a eficácia do modelo em um valor muito próximo à um.

Figura 14: Relação entre acurácia do modelo e número de épocas.

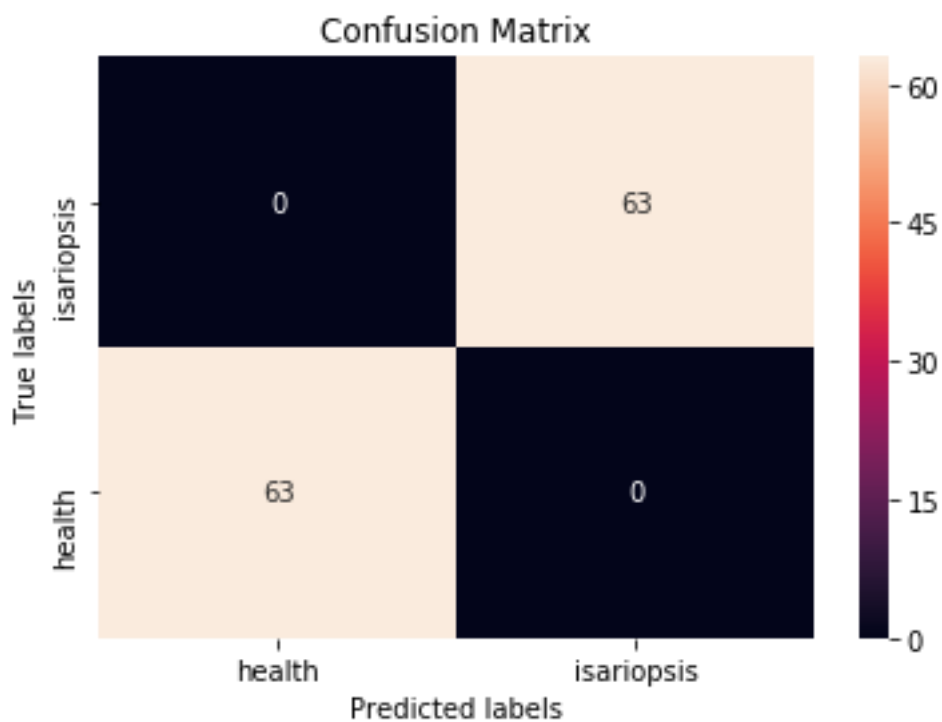


Fonte: Elaborado pelo autor.

Além do conjunto de dados de teste e treino, fez-se uso do conjunto de validação, ele possui a mesma proporção de imagens que o conjunto de testes, e

serve como uma segunda validação aos dados de teste. Na Figura 14 pode-se observar a matriz de confusão e verificar que não houve nenhum falso positivo, todas as 126 imagens do conjunto de validação foram classificadas corretamente.

Figura 15: Matriz de confusão.



Fonte: Elaborada pelo autor.

7 CONCLUSÃO

Neste trabalho foi apresentado o sistema desenvolvido para identificar a presença da doença mancha das folhas em uma videira, utilizando redes neurais convolucionais. O modelo treinado apresentou ótima precisão e taxas de erro próximas de zero. Para identificação de novas doenças, uma das maiores dificuldades esteve em ampliar a cobertura de doenças, através da inserção de novas imagens, isso ocorreu diante das dificuldades em coletar novas imagens.

Foi possível também compreender as principais etapas que compõem uma arquitetura de rede neural convolucional e analisar os parâmetros que envolvem a sua construção, percebendo que grande parte deles estão relacionados, verificando também a importância da etapa de pré-processamento.

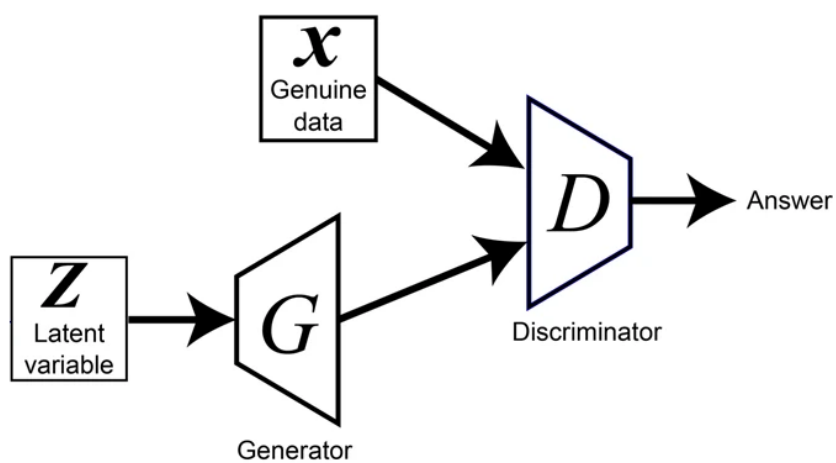
Notou-se também as principais diferenças entre uma das arquiteturas mais populares, VGG16 e a arquitetura utilizada MobileNet, estas diferenças estão nas camadas específicas, porém também no uso da técnica *depthwise separable convolutions*.

7.1 TRABALHOS FUTUROS

Diante das dificuldades em ampliar o banco de dados, principalmente em doenças com rara ocorrência e que afetam drasticamente uma plantação, se vê a necessidade em novas tecnologias para crescer à base de dados. As redes adversariais generativas (GAN's) são um tipo de rede neural que permite gerar dados. Estas redes utilizam dois tipos de arquiteturas, um gerador e outro

discriminador, podendo o discriminador ser uma rede neural convolucional ou até mesmo ser feito manualmente, a visão geral de uma GAN pode ser vista na Figura 14 abaixo.

Figura 16: Visão geral de uma GAN.



Fonte: Naruse, Makoto (2019).

8 REFERÊNCIAS

PLANT VILLAGE. Base de dados de Fitopatologia. Suíça: Escola politécnica federal de Lausana, 2016. Disponível em <<https://github.com/spMohanty/PlantVillage-Dataset>>. Acesso em 10 Jan 2019.

MOHANTY, Sharada Prasanna; HUGHES, David; SALANTHÉ, Marcel. **Using Deep Learning for Image-Based Plant Disease Detection**. Digital Epidemiology Lab, EPFL, 2016.

Gonzalez, Rafael C; Woods, Richard E. **Processamento de Imagens Digitais**, Editora Edgard Blucher, São Paulo, 2000

KINGMA, Diederik, LEI BA, Jimmy. **ADAM: A Method for Stochastic Optimization**. ICLR, 2015.

MOTOHKA, Takeshi;NASAHARA, Kenlo; OGUMA, Hiroyuki; TSUCHIDA, Satoshi. **Applicability of Green-Red Vegetation Index for Remote Sensing of Vegetation Phenology**. Remote Sensing, 2010.

SRIVASTAVA, Nitish; HINTON, Geoffrey; KRIZHEVSKY, Alex; SUSTSKEVER, Ilya; SALAKHUTDINOV, Ruslan. **Dropout: A Simple Way to Prevent Neural Networks from Overfitting**. Journal of Machine Learning Research, 2014.

MIYAZAKI, Caio K. **Redes neurais convolucionais para aprendizagem e reconhecimento de objetos 3D**. Engenharia Elétrica – EESC/USP. São Carlos, 2017.

PIZZI, Daniel A. **Análise da influência de descritores de textura LMP na entrada de redes neurais convolucionais**. Engenharia Elétrica – EESC/USP. São Carlos, 2017.

AMARA, Jihen; BOUAZIZ, Bassem; ALGERGAWY, Alsayed. **A Deep Learning-based Approach for Banana Leaf Diseases Classification**. B. Mitschang (Hrsg). BTW - Workshopband, 2017.

FUENTES, Alvaro; YOON, Sook; KIM, Sang C; PARK, Dong S. **A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition**. MDPI Sensors, 2017.

KAMILARIS, Andreas; PRENAFETA-BOLDÚ, Francesc X. **Deep learning in agriculture: A survey**. Computers and Electronics In Agriculture 147, 2018.

HOWARD, Andrew G; ZHU, Menglong; CHEN, Bo; KALEJECHENKO, Dmitry; WANG, Weijun; WEYWAND, Tobias; ANDREETO, Marco; ADAM, Hartwig. **MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications**. Google Inc, 2017.

SIMONYAN, Karen; ZISSERMAN, Andrew. **VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION**. ICLR, 2015.

BAO, Jianmin; CHENG, Dong; WEN, Fang; LI, Houqianq; HUA, Gang. **CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training**. Computer Vision Foundation, 2015.

KARPATHY, A. **Convolutional Neural Networks for visual recognition**. 2017. Disponível em <<http://cs231n.github.io/convolutional-networks/>>. Acesso em 10 Jan.2019.

ARAÚJO, F. H; CARNEIRO, A. C; SILVA, R. R. **Redes Neurais Convolucionais com tensorflow: Teoria e prática**. III Escola Regional de Informática do Piauí. Livro Anais – Artigos e Minicursos, v. 1, n. 1, p. 390-400, 2017.

BAO, Jianmin; CHENG, Dong; WEN, Fang; LI, Houqianq; HUA, Gang. **CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training**. Computer Vision Foundation, 2015.