

Detecção de patologias em imagens através da transferência de conhecimento utilizando redes MobileNet pré-treinadas

Alexandre Maculan Raisel
Ciência da Computação
Instituto Federal Catarinense
Videira, Santa Catarina
alexandremraisel@gmail.com

Resumo—O presente projeto propõe a utilização de uma rede MobileNet para o desenvolvimento de uma aplicação para detecção de patologias presentes em folhas de videiras. Utilizando a biblioteca do TensorFlow aliado a Api do Keras, foi importado as imagens já padronizadas de uma base dados pública, em seguida, foi necessário efetuar o devido pré-processamento e normalização dos dados antes de treinar a rede, cuja, foi importada o modelo pré-treinado do próprio repositório do TensorFlow. Após congelamento de algumas camadas e definições dos parâmetros da rede, foi efetuado a fase de treino e posteriormente os testes, estes apresentados através da matriz de confusão e de gráficos analíticos, atingindo sucesso com uma taxa de acerto acima de 90%. Ao fim, é explanado resultados e considerações finais sobre o modelo, os dados, e possíveis melhorias.

Palavras-chave—MobileNet, Patologias, Videiras, Transferência de Conhecimento, TensorFlow, Keras

I. INTRODUÇÃO

O uso de tecnologias para o reconhecimento de imagens proporcionou aos computadores a habilidade de captar e interpretar representações de problemas físicos, facilitando o trabalho humano e contribuindo para os avanços em diversas áreas. No setor da agricultura não é diferente, atualmente, há inúmeras aplicações utilizando-se dessa tecnologia, entre os usos mais populares está a detecção de patologias para posteriormente auxiliar na tomada de decisões. O monitoramento e controle dos fatores fitopatogênicos de forma rápida e precisa são cruciais para que se obtenha uma boa produção, independente da cultura.

O reconhecimento de imagens é uma ferramenta que está cada vez mais presente no cotidiano de toda sociedade, desde o início da produção até o consumo final. Trata-se de uma área que engloba a área da visão computacional aliada ao aprendizado de máquina (*machine learning*). O primeiro trata de métodos e teorias para a extração de informações úteis contidas em imagens para que possam ser compreendidas pelo computador [1]. Já o segundo, é uma área da inteligência artificial que evoluiu do reconhecimento de padrões, permitindo aos computadores a possibilidade de aprender sem envolver programação explícita e a partir disso, produzir decisões e resultados [2]. O campo do aprendizado de máquina possui inúmeras abordagens e subáreas, dentre elas, há algumas

especializadas para determinados trabalhos. Do ponto de vista do reconhecimento de imagens, destaca-se um método que se mostrou bastante eficiente a partir de 2012, que são as redes neurais convolucionais, desenvolvidas com base em redes neurais tradicionais e aprendizado profundo (*Deep Learning*), que consiste em treinar um modelo até que este seja capaz de prever resultados com exatidão [3].

Além da eficiência para se trabalhar com imagens, vídeos e outras mídias... As redes neurais convolucionais dispõe de diversas arquiteturas e documentação na Internet, sendo assim, é possível escolher a arquitetura que melhor satisfaz as restrições do problema, sejam elas, de dados, de custo computacional, etc [3].

A arquitetura MobileNet, apresentada em 2017, se baseia em criar um rede neural convolucional que seja pequena e rápida focada em desenvolver um modelo reduzido (em relação as demais arquiteturas de redes convolucionais) mas com boa capacidade de acerto. Esta por sua vez, demanda de menor custo computacional e se torna uma rede de maior facilidade de manipulação (fase treinamento, testes e alterações) [4].

Ainda pensando em atenuar a fase de treinamento das redes neurais (fase mais custosa) e estimular sua taxa de acerto, foi pensado no conceito da transferência de conhecimento para redes pré-treinadas, esse método baseia-se em utilizar uma rede já treinada para determinado trabalho e introduzir novos dados na fase de treinamento [5].

Com base nas tecnologias apresentadas, esse trabalho propõe utilizar a transferência de conhecimento em uma rede MobileNet pré-treinada, para detectar patologias presentes em folhas de videiras. Visto que, ainda não há aplicação desse tipo de modelo para a amostra escolhida.

O método consiste utilizar à biblioteca TensorFlow agregada a biblioteca do Keras para pré-processamento e normalização dos dados, montagem do modelo, treinamento, teste e apresentação dos resultados através da matriz de confusão. Os quais chegaram a taxa de acerto acima de 90%, foi apresentado também na sessão de resultados os gráficos de acurácia e de perda para posterior explanação.

Finalmente, o projeto apresentará as conclusões obtidas do desenvolvimento do projeto, do modelo e dos dados.

II. REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais são recursos de IA inspiradas na estrutura do cérebro humano, mais especificamente nos neurônios biológicos, porém com uma quantidade muito limitada. Inicialmente foi projetada para resolver pequenos problemas, entretanto, com a evolução do método e dos computadores foi possível adicionar mais camadas e parâmetros, caracterizando o aprendizado profundo (*Deep Learning*), viabilizando a resolução de problemas mais complexos e não-lineares, consequentemente, ganhando mais utilidade [6].

Sua estrutura pode ser observada na Figura 1, geralmente é formada por várias unidades de processamento (chamadas de neurônios) organizadas em camadas profundas e conectadas por canais associados a pesos ajustáveis. A habilidade mais importante das redes neurais é extrair características de conforme o nível de abstração das camadas e com isso melhorar seu desempenho [2]

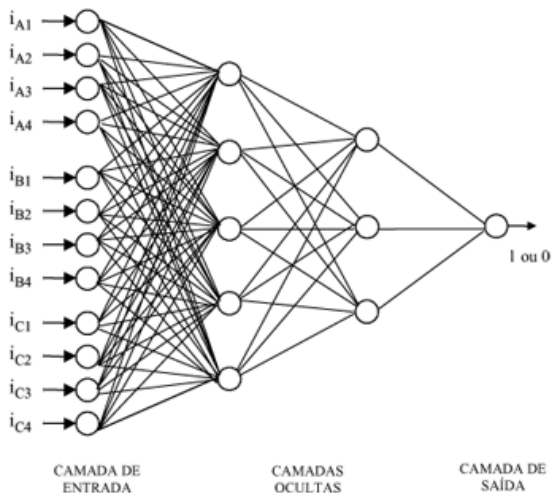


Fig. 1. Representação de uma rede neural artificial [7]

Para uma rede neural artificial ser utilizável, a mesma deve passar por um aprendizado, realizado através de treinamento, o qual é feito por meio de um processo iterativo de ajustes aplicado a seus pesos. O processo de aprendizado termina somente quando essa consegue generalizar soluções para um determinado problema. Todo seu conhecimento aprendido fica armazenado nos pesos atribuídos às conexões entre os neurônios [7].

Contudo, essa tecnologia ainda é muito ineficiente para classificação de grandes quantidades de imagens devido ao alto custo computacional envolvido. [3]

III. REDES NEURAIS CONVOLUCIONAIS

As redes neurais convolucionais surgiram em 1988, quando foi apresentada a primeira aplicação de sucesso por Yann LeCun e seus colaboradores, esta se caracteriza por possuir

pelo menos uma camada de convolução, termo utilizado para definir um operador linear que, sobre duas funções iniciais, resulta em uma terceira resultado da soma do produto dessas funções ao longo de estágios sucessivos. A sua principal vantagem em comparação com as redes tradicionais é para a análise de imagens, por exemplo, é que necessitam de uma menor quantidade de parâmetros em relação a redes do mesmo tamanho que utilizam camadas totalmente conectadas. Ainda que, as camadas internas não são conectadas com todas as outras camadas seguintes, logo, há menor necessidade de processamento para o ajuste de pesos, auxiliando no desempenho do treinamento [3]. Em outras palavras, uma rede neural convolucional segue o pretexto de filtrar linhas, curvas e bordas e em cada camada acrescida transformar essa filtragem em uma imagem mais complexa [8]. Fator esse, que impulsionou a popularidade dessa arquitetura, reduzindo o custo computacional de processamento e, consequentemente, tornando-se mais acessível à comunidade desenvolvedora.

A arquitetura dessa rede pode ser percebida na Figura 2, a qual baseia-se em camadas subdivididas em duas partes principais: extrator de características e classificador. A extração de características é formada sobretudo por filtros (*kernels*), camadas convolucionais e de subamostragem (*pooling*). A segunda parte é formada por camadas totalmente conectadas (*fully-connected*) e tem como objetivo definir o espaço das características onde a separação entre as classes é a melhor possível. A eficácia das redes convolucionais provém do uso de ambas estruturas e do treinamento do sistema em único processo conjunto [10].

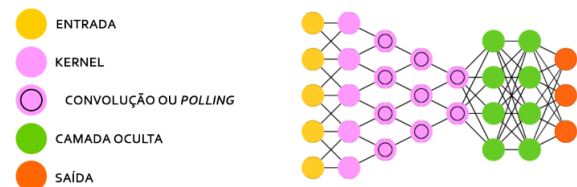


Fig. 2. Estrutura de uma Rede Neural Convolucional [9]

Para que a rede consiga aprender qualquer tipo de funcionalidade, são implementadas as funções de ativação trazendo a não-linearidades ao sistema. Há muitas funções, como sigmoidal ou tangente hiperbólica, embora a mais utilizada seja a *ReLU* (*Rectified Linear Units*) e suas variantes por serem as mais eficientes computacionalmente e sem grandes diferenças de acurácia quando comparada a outras funções [8].

Redes neurais convolucionais muito complexas tendem a gerar modelos igualmente complexos, normalmente, esses modelos são construídos sob forte influência de apenas um subconjunto ou caminho de neurônios em detrimento dos demais. Nesses casos há grandes chances de ocorrer sobreajuste de dados (*overfit*), em outras palavras, ocorre quando o modelo treinado reconhece bem as imagens de treino, mas não tão bem assim as imagens de teste. Para tentar aliviar isso, existem métodos conhecidos como regularizações. Um dos

mais utilizados é o *Dropout* que desconsidera alguns neurônios e força estimular outros caminhos no decorrer do treinamento [11].

Depois de extrair as informações, a rede neural convolucional passa para o processo de classificação. Esta etapa do processo é responsabilidade das camadas totalmente conectadas (*fully-connected*), uma rede neural tradicional. Esta etapa recebe como entrada a saída da última camada convolucional (que são as características aprendidas) e procede na separação entre as categorias do problema. A última camada da rede (camada de saída) é quantificado (0 a 1) a ou as hipóteses da solução entregue [10]. Geralmente, é utilizado a função de ativação *Softmax*, que recebe como entrada um vetor de K números reais e a normaliza em uma distribuição de probabilidade que consiste em K probabilidades proporcionais às exponenciais dos números de entrada [11].

IV. REDES MOBILENET

Embora as redes neurais convolucionais mais potentes apresentem altas taxas de acertos com o devido treinamento, estas ainda necessitam de elevado custo computacional. Por sua vez, desenvolvedores da Google apresentaram em 2017 um novo conceito, uma arquitetura convolucional separada em profundidade (*depthwise separable convolutions*), nomeada de MobileNet, focada para ser um modelo pequeno, rápido e fácil de integrar em plataformas *mobile* [4].

Essa abordagem de convoluções fatorizadas são baseadas em dois componentes. Toda convolução padrão é fatorizada em uma convolução em profundidade e uma convolução 1×1 (nomeada de convolução pontual) [4].

Uma convolução padrão filtra e combina entradas em um novo conjunto de saídas em uma única etapa [12]. Já a convolução separável em profundidade divide isso em duas camadas:

- Camada separada para filtragem: fatora uma convolução padrão e aplica um único filtro para cada canal de entrada.
- Camada separada para combinação: chamada de convolução pontual, aplica uma convolução 1×1 para combinar as saídas da convolução em profundidade.

Essa fatoração tem o efeito de reduzir drasticamente a computação e o tamanho do modelo. A Figura 3 apresenta uma comparação entre a convolução padrão (3 (a)) e as etapas de fatoração e combinação pontual (3 (b)) para criar um filtro separável em profundidade.

Com essa abordagem, o custo computacional das convoluções do modelo MobileNet em razão do custo das convoluções padrões pode ser denotado pela seguinte expressão matemática:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2(1)}$$

V. TRANSFERÊNCIA DE CONHECIMENTO

Com o intuito de reduzir o tempo de treinando e tentar suprir a falta de conjuntos de dados em modelos de aprendizado profundo, pode ser utilizado a transferência conhecimento como

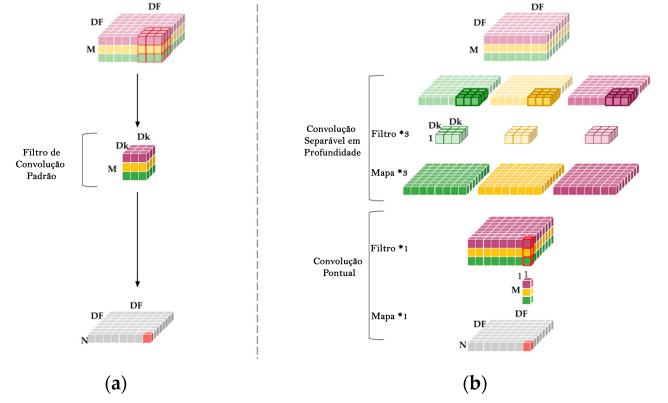


Fig. 3. Comparação de Arquiteturas: Convolução Padrão × Convolução Separável em Profundidade [13]

solução para reciclar o conhecimento de modelos previamente treinados com grande volume de dados. Para a classificação de imagens consiste em utilizar os vetores de características gerados por uma rede convolucional previamente treinada com seus pesos sinápticos ajustados para reconhecer e extrair características de um conjunto de dados para o qual foi treinado, esses vetores de características gerados serão os descritores de cada imagem presente na base de dados. Os vetores gerados, podem ser usados como entrada para um novo classificador para que seja treinado com esses vetores ou podem ser dados como entrada para a camada totalmente conectada de classificação da própria arquitetura que foi utilizada para extração das características [5].

Essa abordagem tem apresentado bons resultados e é uma alternativa muito vantajosa, pois dispensa o tempo que seria gasto treinando uma rede neural robusta do zero para um domínio específico.

VI. METODOLOGIA

Este capítulo propõe a apresentação dos métodos sugeridos para a abordagem do problema, ademais, o presente trabalho classifica-se como, quantitativa, em relação à abordagem do problema e exploratória, em relação aos objetivos.

Foi escolhido o ambiente do *Google Colaboratory*, o qual é um ambiente de *notebooks Jupyter* que não requer configuração e é executado na nuvem e é facilmente integrável as tecnologias utilizadas para o desenvolvimento da aplicação [14].

Foi utilizado a biblioteca do TensorFlow, o qual possui um ecossistema abrangente e flexível de ferramentas, bibliotecas e recursos da comunidade que permite aos desenvolvedores criar e implantar aplicativos com tecnologia de *machine learning* [15]. Já o Keras é uma API, consistente e simples, que minimiza o número de ações do usuário necessárias para casos de uso comuns e fornece mensagens de erro claras e acionáveis [16]. Dessa maneira, foi utilizado o Keras para implementar o modelo, bem como, algumas funcionalidades presentes no TensorFlow.

A. Base de Dados

A base de dados utilizada será uma base pública de imagens de folhas de uvas, divididas em quatro classes, sendo três classes de folhas doentes (doenças: *Black rot* (podridão negra), *Esca Black Measles* (Mancha negra) e *Leaf Blight* (Ferrugem), e uma classe *Healthy* (folhas saudáveis). Ao total somam 3.209 imagens padronizadas, onde a imagem da folha está centralizada sobre um fundo preto, como pode ser observado na figura 4.



Fig. 4. Item exemplo da base de dados

B. Normalização de Dados e Pré-processamento

A normalização dos dados consiste em padronizar as resoluções de entrada para 224×224 pixels (padrão de entrada para o modelo), além de, equilibrar os pesos de cada valor de entrada. Já o pré-processamento é vetorizado as imagens utilizando a biblioteca *OpenCV* (biblioteca projetada para resolver problemas de visão computacional) [17].

Um truque útil para evitar o *overfitting* é usar o aumento de dados. Adicionando dados artificialmente com algumas alterações de rotação, translação, cortes aleatórios, ajuste de brilho... Graças a essas transformações, é possível obter um conjunto de dados maior ($\times 2, \times 3$ em tamanho) e, em seguida, treinar o modelo de uma maneira muito robusta.

A seguir é dividido os dados de teste (20%) e treino (80%), para que a rede não utilize a mesma base para ambas variáveis e consequentemente prejudique o resultado final.

C. Implementação do modelo e treinamento

A fase de extração de característica e classificação é função empenhada pela rede *MobileNet* pré-treinada, o qual foi definido importando o modelo do Keras. O modelo importado foi projetado para receber entradas de 224×224 pixels por 3 canais de cores, alimentado por milhares de dados presentes na base ImageNet. Esse modelo possui um total de 86 camadas convolucionais, sendo elas 13 convoluções com adição de

camadas *ReLU* (função de ativação) e uma camada de *batch-norm* (Técnica para melhorar a velocidade, o desempenho e a estabilidade). A arquitetura da parte totalmente conectada foi definida manualmente, adicionando o recurso de *Flatten* e *Dropout* entre as camadas de neurônios.

A transferência de conhecimento consiste em re-utilizar um modelo que já possui ajustes em seus pesos para classificar determinado material, para treinar a nova rede, deve-se bloquear algumas camadas convolucionais ou até mesmo todas, dependendo do caso. Para esse projeto foi utilizado a estratégia de congelar todas as camadas convolucionais, dado que, a rede foi pré-treinada por inúmeras classes e dados, apresentado a capacidade de generalizar os novos dados que foram inseridos, além disso, os novos dados transferidos são poucos se for comparar com o que rede já conhece.

Após isso, é definido o otimizador, sua função de *fit*, épocas de treino entre outros parâmetros, os valores para esses fatores serão alterados conforme os resultados obtidos. Dado a disposição desses critérios a rede é treinada e pode ser avaliada.

D. Avaliação

Para avaliar adequadamente a assertividade do modelo proposto será implementada técnicas de matriz de confusão além de apresentação de gráficos, visto que a matriz de confusão apresenta boa fidelidade em relação a acurácia geral [18].

A matriz de confusão é uma medição de desempenho para o problema de classificação em que a saída pode ser de duas ou mais classes. Para determinar essa matriz são necessários alguns dados e predições realizadas pelo classificador [19]. Os indicadores analisados através da matriz de confusão são:

- Taxa de Verdadeiros Positivos (TVP): Ocorre quando a classe objeto de busca é prevista corretamente.
- Taxa de Verdadeiros Negativos (TVN): Ocorre quando a classe objeto que não está sendo buscada é prevista corretamente.
- Taxa de Falsos Negativos (TFN): Ocorre quando a classe objeto que não está sendo buscada é prevista incorretamente.
- Taxa de Falsos Positivos (TFP): Ocorre quando a classe objeto de busca é prevista incorretamente.

Por sua vez, a taxa de acurácia (ACC) pode ser explicada pela seguinte fórmula:

$$ACC = \frac{TVP + TVN}{TVP + TVN + TFN + TFP} = \frac{\text{Predições corretas}}{\text{Total de predições}} \quad (2)$$

Outro meio de representar a matriz de confusão é através de estatísticas:

- *Recall*: quão bom meu modelo é para prever positivos, sendo positivo entendido como a classe que se quer prever.
- *Precisão*: o qual bem meu modelo trabalhou.
- *F-score*: oferece um balanço entre a precisão e o *recall* de nosso modelo.

Outro ponto de avaliação é a utilização de gráficos para retratar alguns dados, o que é bastante útil do ponto de vista analítico, uma vez que, retrata como o modelo está se comportando, auxiliando na tomada de decisões acerca de ajustes a serem feitos.

VII. RESULTADOS E DISCUSSÃO

Após alguns testes e alterações nos parâmetros da rede, o modelo apresentou uma boa taxa de acertos. Na figura 5, há a representação de sua matriz de confusão para as quatro classes treinadas (onde cada classe é representada por um índice [0 à 3]), é possível notar que o modelo tem boa quantidade de acertos em relação ao seu prognóstico ("o que é pelo o que a rede acha que é").

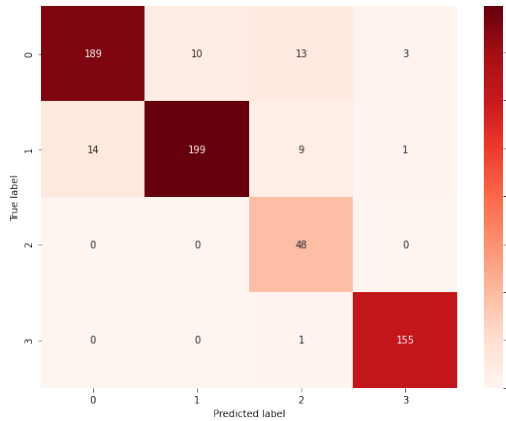


Fig. 5. Matriz de confusão obtida

Estatisticamente, a matriz de confusão pode ser observada na Tabela I, pode-se notar que o modelo obteve uma acurácia média de 92%, a classe que melhor detectou chegou a um valor de 98%, enquanto a pior chegou em 68%, conclui-se que a classe '2' teve piores resultados devido a menor quantidade de dados de treinamento (*support*).

TABLE I
MATRIZ DE CONFUSÃO ESTATÍSTICA

	Precisão	Recall	F1-score	Support
0	0.93	0.88	0.90	215
1	0.95	0.89	0.92	223
2	0.68	1.00	0.81	48
3	0.97	0.99	0.98	156
accuracy			0.92	642
macro avg	0.88	0.94	0.90	642
weighted avg	0.93	0.92	0.92	642

Para acompanhar o comportamento da rede foi plotado os seguintes gráficos, presentes nas Figuras 6 e 7, esses gráficos demonstram o quanto a rede evoluiu ao longo das épocas, é possível notar quais são as respostas do sistema após determinado ajuste de parâmetros, sejam eles, *Dropout*, número de épocas, taxas de aprendizado, adição de camadas ou neurônios, ou até mesmo mudanças nas camadas de extração de características efetuadas pelas camadas convolucionais.

Ainda observando os gráficos, é possível analisar se o modelo está sofrendo ou não de *overfitting*, se fosse o caso, a curva de acurácia sofreria um rápido crescimento e a de perda uma rápida diminuição, logo ambas estagnado. Esse conceito geralmente significa que o modelo chegou a um máximo local e consequentemente, perdendo potencial de melhora. Entende-se que a melhor forma de treinamento de uma rede, gere um gráfico de crescimento linear, ou menos acentuado, uma vez que, explora melhor o potencial de crescimento.

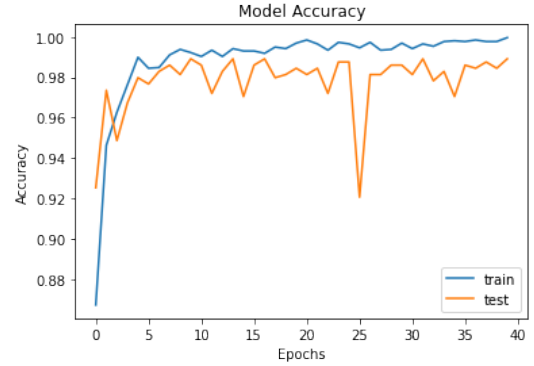


Fig. 6. Acurácia

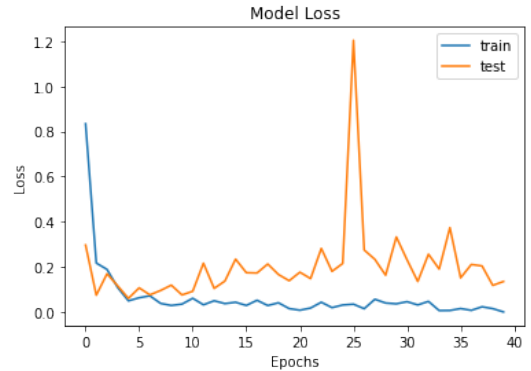


Fig. 7. Perda

VIII. CONSIDERAÇÕES FINAIS

Dada as tecnologias apresentadas e a premissa que o monitoramento e controle dos fatores fitopatogênicos de forma rápida e precisa são cruciais para que se obtenha uma boa produção, esse trabalho propôs a utilização de uma rede MobileNet pré-treinada para a detecção de patologias presentes em folhas de videiras. Visto que, ainda não há aplicação desse tipo de modelo para a amostra escolhida.

O método consiste utilizar a biblioteca TensorFlow agregada a biblioteca do Keras para pré-processamento e normalização dos dados, montagem do modelo, treinamento, teste e apresentação dos resultados através da matriz de confusão, além da demonstração gráfica de comportamento da rede para eventual análise.

O modelo obteve sucesso em seus testes, devido suas altas taxas de acerto apresentadas pela matriz de confusão (acima de 90%), bem como, não sofreu com *overfitting* ou problemas de dados.

Conclui-se que as redes MobileNets aliadas às técnicas de transferência de conhecimentos a redes pré-treinadas são abordagens eficazes, desde que, possuam dados concretos e idôneos, bem como, passem por uma fase de pré-processamento adequando, vale ressaltar que, a normalização e o aumento de dados impacta consideravelmente no modelo.

Como trabalhos futuros, pretende-se melhorar ainda mais os parâmetros da rede, expandir e aumentar a diversidade de classes e dados treinados no modelo, bem como otimizar a rede. Outro ponto de interesse é desenvolver a aplicação MobileNet para plataforma Android, tornando o modelo mais acessível ao usuário.

REFERÊNCIAS

- [1] RUDEK, Marcelo; COELHO, Leandro dos Santos; JUNIOR, Osiris Canciglieri. Visão computacional aplicada a sistemas produtivos: Fundamentos e estudo de caso. Pontifícia Universidade Católica do Paraná, p. 8, 2016.
- [2] KRUG, Alisson et al. Análise e reconhecimento de padrões usando processamento de imagem inteligência artificial. Revista de Iniciação Científica da ULBRA, 01 2008.
- [3] NEUMANN, Bruno Germano. IDENTIFICAÇÃO DE DOENÇAS NA SOJA UTILIZANDO INTELIGÊNCIA ARTIFICIAL POR MEIO DE ANÁLISE DE IMAGENS. 114 p. Monografia (Tecnólogo) — Sistemas de Informação, UNIVERSIDADE DO VALE DO ITAÚNA - Univates, Lageado, 2019.
- [4] HOWARD, Andrew G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017. Disponível em: <http://arxiv.org/abs/1704.04861>.
- [5] NETO, Décio Gonçalves de Aguiar. Transferência de conhecimento utilizando aprendizado profundo para classificação de imagens histopatológicas. 32 p. Monografia (Bacharelado) — Bacharelado em ciência da computação, Universidade Federal do Ceará, Quixadá, 2017.
- [6] FINOCCHIO, Marco Antonio Ferreira. NOÇÕES DE REDES NEURAIS ARTIFICIAIS. CAMPUS CORNÉLIO PROCÓPIO: UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, 2014
- [7] SEGATTO, Ênio Carlos; COURY, Denis Vinicius. Redes neurais aplicadas a relés diferenciais para transformadores de potência. Sba: Controle & Automação Sociedade Brasileira de Automatica, scielo, v. 19, p. 93 – 106, 03 2008. ISSN 0103-1759. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592008000100009&nrm=iso.
- [8] ALVES, Gisely. Entendendo Redes Convolucionais (CNNs). 2018. <https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>.
- [9] VEEN, Fjodor Van. O zoológico da Rede Neural. 2016. <https://www.asimovinstitute.org/neural-network-zoo>.
- [10] DEON, Lucas. DETECÇÃO DE PATOLOGIAS EM FOLHAS DE PLANTAS UTILIZANDO TRANSFER LEARNING. 62 p. Monografia (Bacharelado) — Ciências da Computação, Instituto Federal Catarinense, Videira, 2019.
- [11] VALENTE, Thales. Redes Neurais Convolucionais. 2019. <https://colab.research.google.com/drive/1pbbIqPmZiLoFKDYvmZiF4tHNspTmLi42#scrollTo=ToQxgBUZDOO2>.
- [12] CULFAZ, Ferhat. Transferir aprendizado usando o Mobilenet e o Keras. 2018. <https://towardsdatascience.com/transfer-learning-using-mobilenet-and-keras-c75daf7ff299>.
- [13] CHEN, Lun-Chi et al. Video-based parking occupancy detection for smart control system. Applied Sciences, v. 10, n. 3, 2020. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/10/3/1079>.
- [14] COLAB, What is Colaboratory?. 2020. Disponível em: <https://colab.research.google.com/notebooks/welcome.ipynb>.
- [15] TENSORFLOW, Uma plataforma completa de código aberto para machine learning. 2020. Disponível em: <https://www.tensorflow.org/?hl=pt-br>.
- [16] KERAS, KERAS. 2020. Disponível em: <https://keras.io/>.
- [17] SKVARK, opencv-python 4.4.0.42. 2020. Disponível em: <https://pypi.org/project/opencv-python/>.
- [18] MARQUES, Victor Gutemberg Oliveira. Avaliação do desempenho das redes neurais convolucionais na detecção de ovos de esquistossomose. 2017. https://www.cin.ufpe.br/~tg/2017-1/vgom_tg.pdf.
- [19] SOUZA, Emanuel G de. Entendendo o que é Matriz de Confusão com Python. 2019. <https://medium.com/data-hackers/entendendo-o-que-%C3%A9-matriz-de-confus%C3%A3o-com-python-114e683ec509>.