http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2022.v14.n1.e383
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 28/06/2021 Revisado: 31/01/2022 Aceito: 22/03/2022

DETECÇÃO E RECONHECIMENTO DE PLANTAS DE PEQUENO PORTE UTILIZANDO APRENDIZAGEM DE MÁQUINA

SMALL PLANTS DETECTION AND RECOGNITION USING MACHINE LEARNING

Thales Santos Verne[1], Francisco Assis da Silva[1], Leandro Luiz de Almeida[1], Danillo Roberto Pereira[1], Almir Olivette Artero[2]

[1] Universidade do Oeste Paulista – UNOESTE, Faculdade de Informática de Presidente Prudente, Presidente Prudente, SP.
[2] Universidade Estadual Paulista – UNESP, Faculdade de Ciências e Tecnologia, Departamento de Matemática e Computação, Presidente Prudente, SP.
E-mail: thales4trabalho@hotmail.com, chico@unoeste.br, llalmeida@unoeste.br, almir@fct.unesp.br

RESUMO – A detecção e reconhecimento de plantas sempre foi uma tarefa difícil até mesmo para conhecedores e estudiosos, devido a vasta variedade de plantas encontradas ao redor do mundo. Com o avanço da tecnologia torna-se possível resolver esse problema computacionalmente. Neste trabalho, é apresentado um método para realizar a detecção e reconhecimento de plantas a partir de imagens utilizando algoritmos de visão computacional e inteligência artificial. Os resultados mostram que o custo computacional e a taxa de reconhecimento foram satisfatórios para uso em ambientes controlados. O tempo de processamento para reconhecer cada planta foi de 375 milissegundos, com acurácia de 92%.
Palavras-chave: CNN; Detecção e Reconhecimento de plantas; Rede neural; Aprendizagem de máquina.

ABSTRACT – The detection and recognition of plants has always been a difficult task even for connoisseurs and scholars due to the wide variety of plants found worldwide. With the advancement of technology, it has become possible to solve this problem computationally. This paper presents a method to perform plant detection and recognition from images using computer vision and artificial intelligence algorithms. The results show that the computational cost and recognition rate were satisfactory for use in controlled environments. The processing time to recognize each plant was 375 milliseconds, with an accuracy of 92%.
Keywords: CNN; Plants Detection and Recognition; Neural Network; Machine Learning.

1. INTRODUÇÃO

Ter conhecimento sobre espécie, uso e localização de plantas é de suma importância para o desenvolvimento sustentável da agricultura e da conservação da biodiversidade. Identificar uma espécie de planta é uma atividade dificultosa para o público em geral, e as vezes se torna uma tarefa suscetível a erros, até mesmo para os profissionais da área. Unir conhecimento para identificar plantas a partir de imagem

digitais, é considerado uma solução promissora (JOLY *et al*., 2014).

A utilização de aprendizagem de máquina nos últimos anos cresceu muito devido a suas diversas aplicações em reconhecimento de imagens, voz e texto, principalmente quando se envolve grande volume de dados (INOVA, 2017).

Técnicas de análise de imagens surgiram para automatizar o processo de monitoramento e classificação das plantas, uma delas é a técnica baseada em cores. Entretanto, não é confiável, levando em conta que estão expostas ao clima, influenciando diretamente na extração de características da imagem (YALCIN; RAZAVI, 2016).

Utilizar-se do formato das folhas é uma maneira promissora de se identificar uma planta, pelo fato das folhas serem mais acessíveis e abundantes, se comparado com outras estruturas, como flor, casca ou fruta (ZHANG; LIU, 2013).

Automatizar o processo de identificação das plantas, não é algo trivial, uma vez que as imagens de entrada podem ter características diversas, influenciadas pelos fatores de variação de iluminação, movimento das folhas pelo vento, oscilação da câmera, mudança de zoom, borrões, diferentes perspectiva, entre outros (YALCIN *et al*., 2016) (ZHANG; LIU, 2013).

Em alguns trabalhos, como o de Jamil *et al*. (2015), foram utilizadas técnicas processamento de imagens para a extração da forma, textura e cor, e algoritmos de visão computacional para o reconhecimento da folha.

Este trabalho busca contribuir com uma solução computacional, fazendo o uso de algoritmos de processamento de imagens (mudança de iluminação, cor, zoom, espelhamento, rotação e deslocamento) e técnicas aprendizagem de máquina, para realizar a identificação (detecção e reconhecimento) de algumas espécies de plantas em vasos. É considerado, neste trabalho, realizar o reconhecimento da planta inteira, não somente usando folhas, flores ou alguma parte da planta que tenha de característica mais detalhadas.

Primeiramente, foi montada a base de imagens utilizada neste trabalho, que foi feito capturando imagens em um ambiente controlado com fundo de cor única. Essas imagens passaram por um pré-processamento para diminuição da resolução espacial, e com isso reduzir o custo computacional. Posteriormente, as imagens passaram pelo método de aumento de dados.

Para realizar o reconhecimento das plantas foi utilizada uma CNN (*Convolutional Neural Network*), que possibilita obter uma alta precisão no reconhecimento de objetos em uma imagem. Com o uso de CNN, segundo Yalcin e Razavi (2016), os problemas comumente encontrados são reduzidos, pois ela pode ser utilizada com mudanças de iluminação, foco, cores, perspectiva e tamanho.

A rede CNN atingiu uma acurácia de 81,18% no conjunto de treinamento e 92,19% no conjunto de validação. No experimento de uso dos pesos da rede, o tempo de processamento para cada imagem foi de somente 375 milissegundos.

Após esta seção de introdução, o trabalho está organizado da seguinte maneira. Na Seção 2 são descritos os conceitos a respeito de Redes Neurais Convolucionais. Na Seção 3 é apresentado o método proposto para resolver o problema de detecção e reconhecimento de plantas de pequeno porte a partir de imagens usando CNN. Por fim, na Seção 4 são feitas as considerações finais e propostas de trabalhos futuros

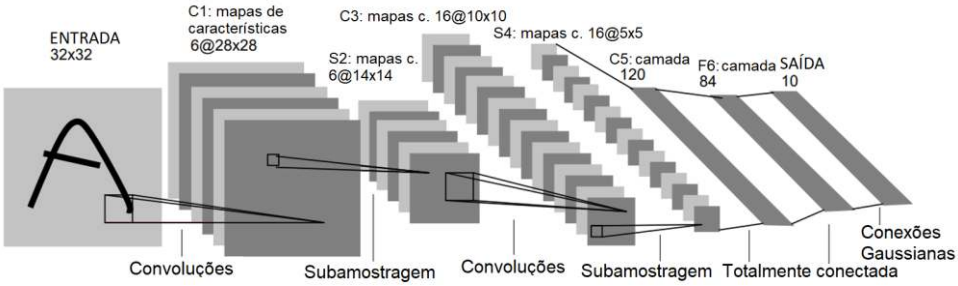
2. REDE NEURAL CONVOLUCIONAL

Uma rede neural convolucional, ou CNN (*Convolutional Neural Network*) é uma rede neural de aprendizado profundo (KRIZHEVSKY; SUTSKEVER; HINTON, 2017), projetada para processar matrizes estruturadas de dados, como imagens. As redes neurais convolucionais são amplamente utilizadas na visão computacional e se tornaram úteis para aplicações visuais, como classificação de imagens, e também obtiveram sucesso no processamento de linguagem natural para classificação de texto. Uma rede neural convolucional simples que auxilia na compreensão dos princípios básicos deste trabalho é a rede neural convolucional LeNet-5. Na Figura 1 tem-se, como exemplo, a arquitetura da LeNet com uma imagem de entrada de um caractere escrito à mão (LECUN *et al*., 1998).

As redes neurais convolucionais são muito utilizadas em captar padrões na imagem de entrada, como linhas, gradientes, círculos ou mesmo olhos e rostos. É essa propriedade que torna as redes neurais convolucionais poderosas para a visão computacional. Ao contrário dos algoritmos de visão computacional anteriores, MLPs (*multilayer perceptron*) e RNNs (*recurrent neural network*), as redes neurais convolucionais podem operar diretamente em na imagem de

entrada e não precisam de nenhum pré-processamento. Uma rede neural convolucional é uma rede neural *feed-forward*, geralmente, com até 20 ou 30 camadas. As redes neurais convolucionais contêm muitas camadas convolucionais empilhadas umas sobre as outras, cada uma capaz de reconhecer formas mais sofisticadas, seguidas por camadas de ativação. Com três ou quatro camadas convolucionais é possível, por exemplo, reconhecer dígitos escritos à mão e com 25 camadas é possível, por exemplo, distinguir rostos humanos (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 1. Arquitetura da LeNet.



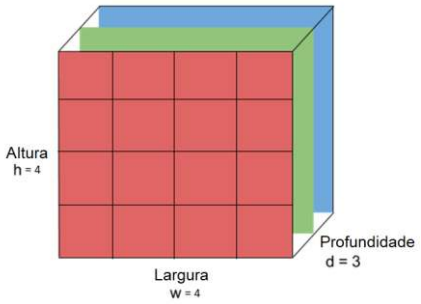
Fonte: Adaptado de Lecun *et al*., (1998).

2.1 Camada Convolucional

Uma camada convolucional contém um conjunto de filtros cujos parâmetros precisam ser aprendidos. A altura e o peso dos filtros são menores do que os do volume de entrada. Cada filtro é convolucionado com o volume de entrada para calcular um mapa de ativação feito de neurônios. Em outras palavras, o filtro é deslizado ao longo da largura e altura da entrada e os produtos escalares entre a entrada e o filtro são calculados em cada posição espacial. O volume de saída da camada convolucional é obtido empilhando os mapas de ativação de todos os filtros ao longo da dimensão de profundidade (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Uma imagem qualquer pode ser definida com 2 valores, sua largura (*width*) e altura (*height*), neste exemplo 4 pixels, com uma profundidade (*depth*) de pixels de 3, sendo eles, RGB (*red, green, blue*), formando assim uma imagem 4x4x3, como ilustrado na Figura 2.

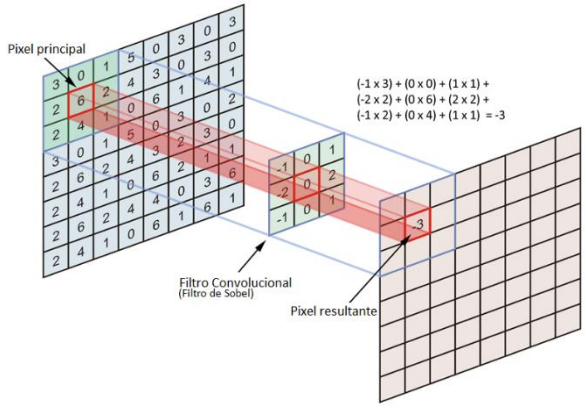
Figura 2. Ilustração de uma imagem com 3 canais de cores.



Fonte: Adaptado de Ulindala (2020).

Uma vez que a largura e a altura de cada filtro são projetadas para serem menores do que a entrada, cada neurônio no mapa de ativação é conectado apenas a uma pequena região local do volume de entrada. Em outras palavras, o tamanho do campo receptivo de cada neurônio é pequeno e igual ao tamanho do filtro. Após o tamaho do filtro ser definido, cada um dos filtros é aplicado a todas as entradas, realizado as convoluções como na pode ser visto na Figura 3 (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Figura 3. Exemplo de uma convolução de um filtro de Sobel em uma entrada.



Fonte: Martins (2020).

Os valores que foram obtidos com a aplicação dos filtros são submetidos a uma função de ativação. A função ReLU (*Rectified Linear Units*) é aplicada à saída de cada camada convolucional e totalmente conectada (LI, 2017),

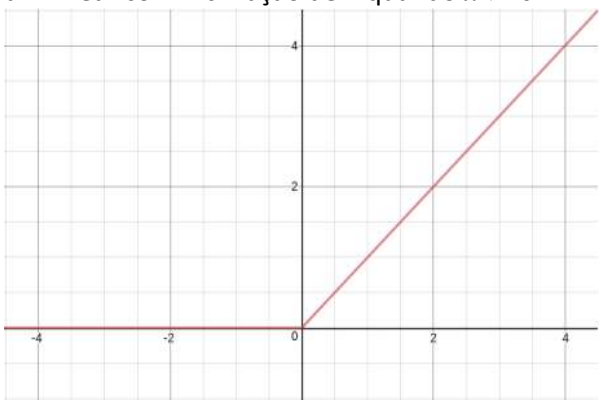
então usa-se os valores ja ativados pela ReLU para aprender o peso que a camada de ativação ReLU irá obter, através do *backpropagation.*

Funciona limitando valores em 0, ou seja,

$$f(x) = \max(0, x) \qquad (1)$$

Simplificando, produz 0 quando $x < 0$, e inversamente, é produzida uma função linear quando $x \geq 0$, ilustrada na Figura 4 (AGARAP, 2018). Há ganhos significativos, pois valores negativos serão convertidos para 0, e o neurônio correspondente não será ativado.

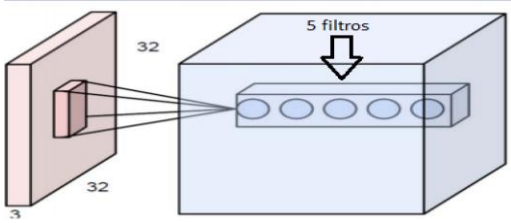
Figura 4. A função de ativação ReLU produz 0 como uma saída quando $x < 0$, e então produz um linear com inclinação de 1 quando $x > 0$.



Fonte: Os autores.

Cada filtro corresponde a uma profundidade (*depth*) como na Figura 5 (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

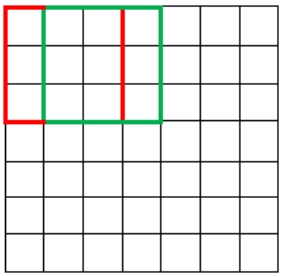
Figura 5. Exemplo de 5 filtros (*depths*) sendo aplicados à mesma região.



Fonte: Albawi, Mohammed e Al-Zawi (2017).

Para percorrer a imagem, é utilizado o *Stride*, que é responsável por definir o tamanhos do deslocamento dos filtros pela entrada a qual é ilustrada na Figura 6 (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Figura 6. Exemplo de deslocamento de *stride* com valor em 1 .



Fonte: DeepAI (2021).

Uma das desvantagens da convolução é a perda de informações que podem existir na borda da imagem. Como os valores só serão capturados quando o filtro deslizar, eles nunca iriam ser processados. Um método muito simples e eficiente para resolver isso é se utilizar do preencimento de zero (*Zero-padding*) como na Figura 7, pois possibilita o filtro a acessar os dados contidos nas estremidades da imagem de entrada. Também é possível com o preenchimento de zeros ter controle do tamanho de saida (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Figura 7. Ilustração de uma imagem com o Zero-Padding aplicado com tamanho 1.

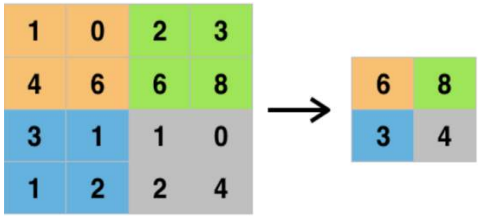
0	0	0	0	0	0	0
0	2	0	1	1	3	0
0	1	0	1	3	0	0
0	1	2	3	3	3	0
0	3	0	0	0	2	0
0	1	0	2	3	3	0
0	0	0	0	0	0	0

Fonte: Os autores.

2.2 Camada de *Pooling*

A Funcionalidade do *Pooling* é reduzir a amostragem final, reduzindo a complexidade das outras camadas, em imagens pode ser semelhantes a reduzir a resolução. O *Pooling* nao afeta o número dos filtros que serão aplicados. *Max-pooling* é um dos tipos mais comuns de métodos de agrupamento. Ele divide a imagem em uma sub-região de retângulos e só retorna o valor máximo dentro daquela sub-região, como na Figura 8 (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Figura 8. Exemplo da saída após aplicar o *Max-pooling*.

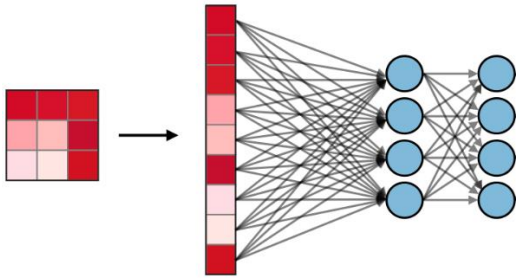


Fonte: Os autores.

2.3 Camada Totalmente Conectada

A camada totalmente conectada (*Fully-connected layer*) é semelhante a como os neurônios são organizados em uma rede neural tradicional, cada nó em uma camada está diretamente conectada com a sua camada anterior e sua próxima camada (ALBAWI; MOHAMMED; AL-ZAWI, 2017) como na Figura 9.

Figura 9. Ilustração de uma camada totalmente conectada (*Fully-connected layer*).



Fonte: AMIDI e AMIDI (2021).

A principal desvantagem desta camada, é que ela possui muitos parâmetros que necessitam de cálculos computacionais complexos (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Na última camada é aplicada a função *softmax* como uma função de ativação (GOODFELLOW; BENGIO; COURVILLE, 2016), definida pela Equação 2.

$$softmax(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \qquad (2)$$

2.4 *Backpropagation*

O termo retro propagação (*backpropagation*) é frequentemente mal interpretado como se ele fosse todo o algoritmo de aprendizagem. *Backpropagation* somente se refere ao método para calcular o gradiente, enquanto outro algoritmo se utiliza desses valores para aprender. Os pesos inicias são aleatórios, e ao decorrer do processamento, estes valores são ajustados para otimizar a acurácia da classificação dos objetos. Devido ao ajuste que é realizado, o erro obtido pela rede é cada vez menor, sempre que a mesma imagem é processada como entrada (GOODFELLOW; BENGIO; COURVILLE, 2016).

3. MÉTODO PROPOSTO

Nesta seção são descritos todos os recursos utilizados, a rede utilizada e os resultados obtidos.

3.1. Base de dados

Todo o conjunto de treinamento, validação e testes, são de plantas que foram obtidas por uma câmera de 12 Megapixels, a partir de um *smartphone* Redmi note 8.

As imagens foram capturadas em um ambiente controlado, utilizando uma bancada com fundo de cor única, porém, com diferentes pontos de vista e iluminação variada para montar a base de dados, como na Figura 10.

Figura 10. Amostra das imagens da base de dados.



Fonte: Os autores.

Foram obtidas ao todo 1.050 fotos, sendo separadas em 10 classes de plantas (105 fotos para cada classe), apresentadas na Tabela 1. As imagens foram separadas da seguinte forma: 750 para treino, 250 para validação da rede e 50 para testes. Todas as imagens utilizadas estão disponíveis no repositório: https://drive.google.com/drive/folders/1w6pLESJJad1gcRmBy6HhBL0_k1P2JRtE.

Tabela 1. Classes de plantas utilizadas.

Nome popular	Nome Científico	Figura 11
Babosa	Aloe Vera	a)
Acácia-branca	Moringa oleifera	b)
Mini jaca	Artocarpus Heterophyllus	c)
Abacate margarida	Persea americana	d)
Pitaia amarela	Selenicereus megalanthus	e)
Caju-anão	Anacardium occidentale	f)
Jasmim-dos-poetas	Jasminum polyanthum	g)
Ora-pro-nóbis	Pereskia aculeata	h)
Mini cacto monstro	Cereus peruvianus monstrosus minima	i)
Flor-da-fortuna	Kalanchoe brasiliensis Cambess	j)

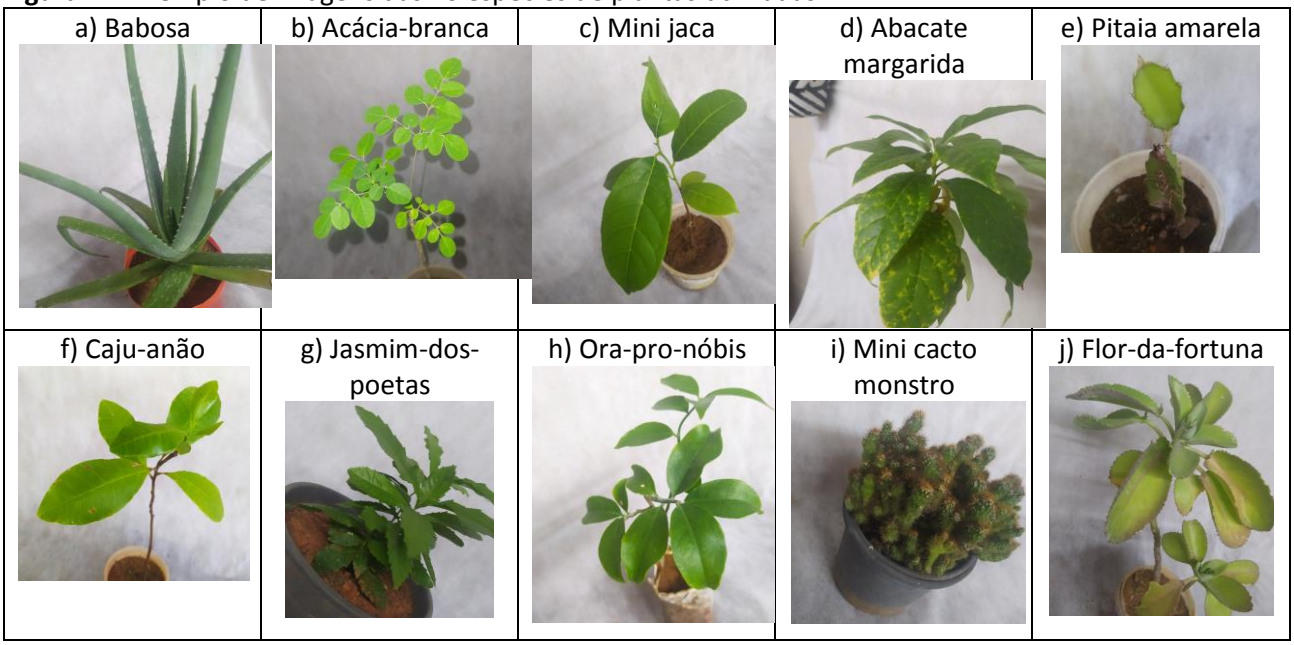
Fonte: Os autores.

Na Figura 11 são apresentadas exemplo de imagens das 10 espécies de plantas utilizada neste trabalho.

3.2. Pré-processamento

As imagens obtidas com o uso da câmera do *smartphone* possuem uma resolução de 3.000x4.000 *pixels*. Houve a necessidade de realizar uma redução na resolução das imagens para reduzir o custo computacional, sendo redimensionadas para a resolução de 500x666 *pixels*.

Figura 11. Exemplo de imagens das 10 espécies de plantas utilizadas.



Fonte: Os autores.

3.3. *Data augmentation*

O aumento de dados (*Data Augmentation*) é um método muito poderoso de se conseguir aumentar a base dados atual sinteticamente utilizando processamento de imagem. Os dados aumentados representam um conjunto mais abrangente de possíveis variações da mesma entrada, neste caso, uma imagem,

minimizando assim a distância entre o conjunto de treinamento e validação, bem como quaisquer conjuntos de testes futuros (SHORTEN; KHOSHGOFTAAR, 2019). Métodos de aumento de dados, como GANs (*Generative Adversarial Networks)* (RICHARDSON; WEISS, 2019) podem "simular" alterações nas imagens de modo que tenham uma melhor compreensão delas (SHORTEN; KHOSHGOFTAAR, 2019).

O método mais fácil e comum para reduzir o sobreajuste (*overfitting*) nos dados da imagem é aumentar artificialmente o conjunto de dados usando transformações, os quais permitem que imagens transformadas sejam produzidas a partir das imagens originais com muito pouco cálculo. Portanto, as imagens transformadas não precisam ser armazenadas no disco. Na implementação, as imagens transformadas são geradas por processamento de imagem do Keras (KERAS, 2021). Os métodos utilizados neste trabalho foram: espelhamento horizontal e vertical, redução no brilho de 20% a 80%, rotação da imagem em até 90º, deslocamento no eixo X, deslocamento no eixo Y, retirar zoom em até 20% e aplicar zoom em até 30%. Esse processo de aumento de dados permitiu expandir o tamanho da base de dados de treinamento de 750 para 6.000 imagens.

3.4. Modelo utilizado

Há uma gama de modelos de redes neurais convolucionais, onde seu principal objetivo é a classificação, detecção ou segmentação dos objetos. Suas diferenças estão na estrutura: número de camadas convolucionais, funções de ativação e tamanho dos filtros. Esses valores impactam diretamente no custo computacional da CNN. Levando isso em consideração, foi utilizada o modelo de rede VGG-16, que é uma versão reduzida de uma CNN. Essa versão foi escolhida por possuir baixo custo computacional e também oferecer uma boa acurácia diante de outras redes neurais apresentadas na Tabela 2.

Tabela 2. Classificação das redes neurais.

CNN	Accuracy	Model Size	Training Time
AlexNet	0.894	217 MB	1140.15s
GoogLeNet	0.898	47.1 MB	332.228s
VGG16	0.905	537.2 MB	1960.2s
VGG19	0.903	558.4 MB	5411.31s
ResNet-50	0.901	94.3 MB	2101.19s
Inceptionv2	0.903	45.1 MB	2187.3s
Inceptionv3	0.901	87.4 MB	6438.72s
Inceptionv4	0.89	165 MB	5787.9s
SENet	0.875	220.9 MB	1794.78s

Fonte: HANG *et al*. (2019).

3.5. Treinamento da rede neural

Para o treinamento da rede neural foi utilizada a API Keras, juntamente com uma biblioteca de código aberto Tensorflow (TENSORFLOW, 2021), que são utilizados para criar e treinar modelos de aprendizagem profunda (*deep learning*). Keras é uma interface de alto nível, acessível e altamente produtiva para resolver problemas de aprendizagem de máquina, com foco em aprendizado profundo moderno (KERAS, 2021).

Todo o processo de aprendizagem foi realizado na nuvem da GOOGLE, o Google Colaboratory PRO, com ambiente configurado em GPU, a NVIDIA Tesla P100 PCIe 16 GB e 25 GB de memória RAM.

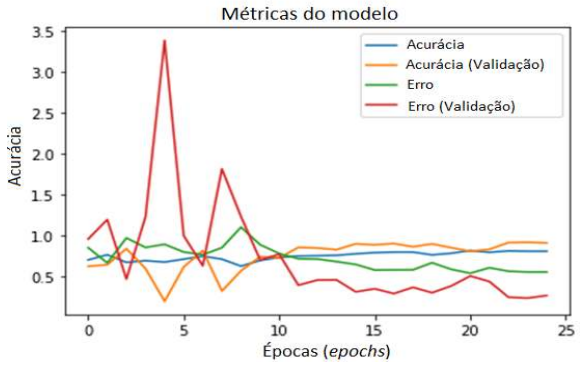
As imagens da base de dados foram utilizadas como parâmento de entrada para alimentar a rede neural. Na configuração da rede, alguns parâmetros são necessários serem informados para a rede, como: taxa de aprendizagem (*Learning rate*) de 0.001, tamanho do lote (*Batch size*) de 8, número de épocas (*epochs*) de 400. No treinamento, existem quatro variáveis, sendo elas, *loss* para o erro no treinamento, o *val_loss* para o erro na validação, *accuracy* para a acurácia no treinamento e *val_accuracy* para acurácia na validação. Quanto mais rápido os valores de erros diminuírem e os de acurácia subirem, melhor é a rede neural, pois tem uma taxa de aprendizagem rápida. Mas, caso a rede demore para convergir para um resultado com boa acurácia, ela conta com uma funcionalidade, a redução da taxa de aprendizagem (*ReduceLROnPlateau*). Nesse caso, a CNN, automaticamente irá reduzir a taxa de

aprendizagem gradativamente, para diminuir a confusão no treinamento e melhorar sua métrica.

O treinamento da rede foi interrompido após atingir a precisão de 92,19% no conjunto de validação, com um tempo total de execução de 27 horas de treinamento, e com 400 épocas.

No final do treinamento, a rede atingiu uma acurácia de 81,18% no conjunto de treinamento e 92,19% no conjunto de validação. Utilizando o Tensorboard do Tensorflow, é possível ver as últimas 25 épocas da rede, a evolução da acurácia e a redução da taxa de erro, como na Figura 12.

Figura 12. Resultado final das métricas da CNN, geradas pelo Tensorboard.



Fonte: Os autores.

3.6. Avaliação dos Resultados

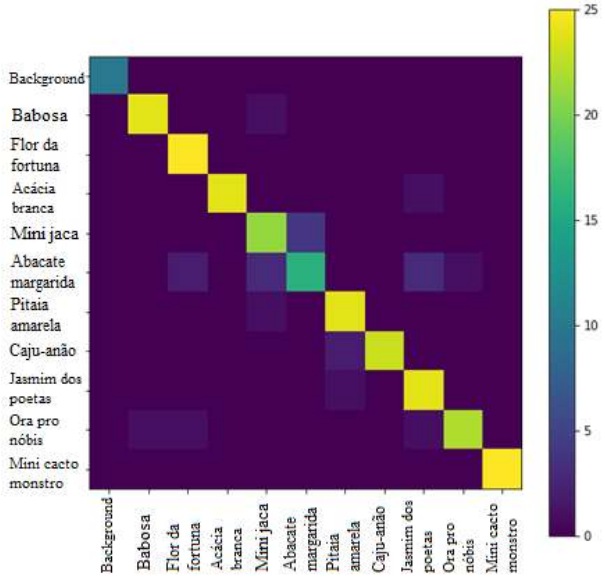
A Tabela 3 contém a precisão de cada classe processada no conjunto de validação. A matriz de confusão ilustrada na Figura 13, representa que a rede neural desempenhou um trabalho satisfatório na sua aprendizagem, mesmo possuindo alguns pontos isolados, que representam a confusão da rede entre as amostras na validação. Os resultados foram satisfatórios, atingindo 81,18% no conjunto de treinamento e 92,19% no conjunto de validação. A partir desses resultados é possível notar que se pode reconhecer uma planta em um ambiente controlado, como o usado neste trabalho com fundo de cor única, sem a necessidade de se ter detalhes específicos da folha, casca, fruto ou flor, como em outros trabalhos.

Tabela 3. Classificação das 10 espécies de plantas utilizadas.

	Precisão	**Conjunto de dados**
Babosa	96%	25
Flor-da-fortuna	89%	25
Mini jaca	81%	25
Abacate margarida	80%	25
Pitaia amarela	89%	25
Caju-anão	100%	25
Jasmim-dos-poetas	83%	25
Ora-pro-nóbis	96%	25
Mini cacto monstro	100%	25
Acácia-branca	100%	25

Fonte: Os autores.

Figura 13. Matriz de confusão da validação da rede.



Fonte: Os autores.

Com as imagens da base de testes (imagens que não foram utilizadas em nenhuma outra etapa, seja ela de treinamento ou validação) foi realizado um experimento com o objetivo de se medir o desempenho de uma aplicação real fazendo o uso do treinamento da rede. Primeiramente, foram sorteadas 40 das 50 imagens disponíveis. Essas imagens foram submetidas a um algoritmo que utiliza os pesos obtidos da rede treinada para o reconhecimento. Os resultados mostraram que o tempo de execução total foi de aproximadamente 15 segundos, sendo 375 milissegundos para cada imagem.

4. CONSIDERAÇÕES FINAIS

Tendo em vista o resultado obtido, a metodologia mostra-se promissora em cenários que demandam resposta rápida e com acurácia, sendo seu tempo de processamento para cada imagem de somente 375 milissegundos e acurácia de 92,5%.

É possível diminuir o tempo de treinamento com a redução do tamanho da imagem. Entretanto, pensa-se que isso iria implicar na perda de acurácia, pois detalhes da planta serão perdidos na redução de escala.

Para trabalhos futuros, acredita-se que uma maior base de dados, uso de redes neurais mais complexas e o refinamento dos parâmetros utilizados, pode propiciar melhores resultados com redução de custo computacional e de tempo reduzido, pois será necessária uma menor quantidade de épocas para a rede neural obter resultados semelhantes ou melhores, devido a extração de mais características da planta, que implicam diretamente na acurácia da CNN

Modelos superiores ao deste trabalho podem ser levados até aplicações em *smartphones* por terem características de boa acurácia e baixo custo computacional, com isso, poderia auxiliar um usuário no seu dia a dia a identificar uma planta (que fosse previamente treinada), utilizando a própria câmera do dispositivo.

REFERÊNCIAS

AGARAP, A. F. M. **Deep Learning using Rectified Linear Units (ReLU).** 2018. Disponível em: https://arxiv.org/pdf/1803.08375.pdf. Acesso em: 15 jun. 2021.

AMIDI, A.; AMIDI, S. Convolutional Neural Networks cheatsheet. Disponível em: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks. Acesso em: 20 jun. 2021.

ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. 2017. *In*: INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY (ICET). https://doi.org/10.1109/ICEngTechnol.2017.8308186.

DEEPAI. What is Stride (Machine Learning)?. Disponível em: https://deepai.org/machine-learning-glossary-and-terms/stride#:~:text=Stride%20is%20a%20parameter%20of,or%20unit%2C%20at%20a%20time. Acesso em: 15 jun. 2021.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep learning. 2016. DOI: https://doi.org/10.4258/hir.2016.22.4.351. Disponível em: https://e-hir.org/journal/view.php?id=10.4258/hir.2016.22.4.351. Acesso em: 15 jun. 2021.

HANG, J.; ZHANG, D.; CHEN, P.; ZHANG, J.; WANG, B. **Classification of Plant Leaf Diseases Based on Improved Convolutional Neural Network**. 2019. DOI: https://doi.org/10.3390/s19194161.

INOVA. AGÊNCIA DE INOVAÇÃO DA UNICAMP. 2017. **Deep learning é tecnologia de aprendizado de máquina que mais cresce em todo o mundo**. Disponível em: https://www.inova.unicamp.br/noticia/deep-learning-e-tecnologia-de-aprendizado-de-maquina-que-mais-cresce-em-todo-o-mundo. Acesso em: 15 jun. 2021.

JAMIL, N.; HUSSIN, N. A. C.; NORDIN, S.; AWANG, K. Automatic Plant Identification: Is Shape the Key Feature?. *In*: IEEE INTERNATIONAL SYMPOSIUM ON ROBOTICS AND INTELLIGENT SENSORS (IEEE IRIS2015). 2015. DOI: https://doi.org/10.1016/j.procs.2015.12.287. Disponível em: https://www.sciencedirect.com/science/article/pii/S1877050915037886. Acesso em: 10 jun.2021.

JOLY, A.; GOËAU, H.; BONNET, P.; BAKIĆ, V.; BARBE, J.; SELMI, S.; YAHIAOUI, I.; CARRÉ, J.; MOUYSSET, E.; MOLINO, J.; BOUJEMAA, N.; BARTHÉLÉMY, D. **Interactive plant identification based on social image data**, 2014. Disponível em: https://www.sciencedirect.com/science/article/pii/S157495411300071X. Acesso em: 15 jun. 2021. https://doi.org/10.1016/j.ecoinf.2013.07.006

KERAS. 2021. Disponível em: https://keras.io. Acesso: 15 jun. 2021.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. 2017. DOI: https://doi.org/10.1145/3065386.

LI, Y. **A Convergence Analysis of Two-layer Neural Networks with ReLU Activation**. 2017. Disponível em:

https://arxiv.org/pdf/1705.09886.pdf. Acesso em: 10 jun. 2021. https://doi.org/10.1145/3065386

MARTINS, J. P. Notas Sobre Redes Neurais Convolucionais. 2020. Disponível em: https://jpvmm.github.io/first-post.html. Acesso em: 20 jun. 2021.

RICHARDSON, E.; WEISS, Y. On GANs and GMMs. 2019. Disponível em: https://arxiv.org/abs/1805.12462. Acesso em: 23 jun. 2021.

SHORTEN, C.; KHOSHGOFTAAR, T. A survey on Image Data Augmentation for Deep Learning. 2019. DOI: https://doi.org/10.1186/s40537-019-0197-0. Disponível em: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0. Acesso em: 15 jun. 2021.

TENSORFLOW. 2021. Disponível em: https://www.tensorflow.org. Acesso em: 15 jun. 2021.

ULINDALA, P. R. Convolutional Neural Networks (CNN's) — A practical perspective. 2020. Towards data science. Disponível em: https://towardsdatascience.com/convolutional-neural-networks-cnns-a-practical-perspective-c7b3b2091aa8. Acesso em: 24 jun. 2021.

YALCIN, H.; RAZAVI, S. Plant Classification using Convolutional Neural Networks. *In*: FIFTH INTERNATIONAL CONFERENCE ON AGRO-GEOINFORMATICS (AGRO-GEOINFORMATICS).1., 2016. Tianjin, China. **Anais** […].Tianjin, China, 2016.. https://doi.org/10.1109/Agro-Geoinformatics.2016.7577698

ZHANG, N.; LIU, W. Plant leaf recognition method based on clonal selection algorithm and K nearest neighbor**. Journal of Computer Applications**, 2013. https://doi.org/10.3724/SP.J.1087.2013.02009