

Classificação de doenças em batata baseado em imagens das folhas de batata utilizando *Deep Learning*

Diego R. Biondo ¹ e Roberto S. Inoue ²

Abstract—A batata é o quarto alimento básico mais consumido em todo o mundo, no entanto, as doenças da batata são a principal causa da redução na qualidade e quantidade da colheita. A classificação inadequada e a detecção tardia do tipo de doença pioram drasticamente as condições da planta. Várias doenças, como a requeima e a pinta preta, influenciam significativamente a qualidade e a quantidade das batatas e a interpretação manual destas doenças foliáceas é demorada e complexa, visto que exige um alto nível de especialização. Neste artigo é apresentado um sistema desenvolvido para classificar os tipos de doenças em plantas de batata, com base nas condições da folha, utilizando *Deep Learning* com o modelo de arquitetura de rede neural convolucional (CNN) para obter um sistema de classificação preciso. O conjunto de dados (*dataset*) utilizado para treinar o modelo foi o PlantVillage. Por meio do modelo proposto em questão, obteve-se uma precisão de 92,57% na identificação das doenças.

Palavras-chave - Classificação de doenças em folhas, *Deep Learning*, Rede neural convolucional (CNN), Agricultura, PlantVillage

I. INTRODUÇÃO

A batata é a terceira cultura alimentar mais importante do planeta e a primeira commodity não grão. Estima-se que mais de um bilhão de pessoas consomem batata diariamente no mundo e sua produção mundial anual supera 330 milhões de toneladas em uma área de 18 milhões de hectares. No Brasil, a batata é a hortalíça mais consumida, com uma produção anual de aproximadamente 3,5 milhões de toneladas em uma área de cerca de 130 mil hectares. De acordo com Associação Brasileira da Batata (ABBA), o agronegócio da batata envolve em torno de cinco mil produtores em 30 regiões de sete estados brasileiros (MG, SP, PR, RS, SC, GO e BA). A maior parte da produção nacional é comercializada in natura, sendo apenas 10% destinados ao processamento industrial, nas

formas de pré-frita congelada, chips e batata-palha [1].

No entanto, a batata pode ser afetada por muitas doenças que afetam os estágios pré e pós-colheita da planta [2]. A presença de doenças durante o período de crescimento pode reduzir a qualidade e a quantidade dos produtos agrícolas. As principais causas destas doenças são microrganismos, doenças genéticas e agentes infecciosos como bactérias, fungos e vírus.

Fungos e bactérias são os principais responsáveis pelas doenças da folha da batata. Como exemplos, pode-se citar as doenças fúngicas requeima e a pinta preta [3]. A requeima, causada pelo Oomycota *Phytophthora infestans* de Bary, e a pinta preta, ocasionada pelo fungo *Alternaria solani* Sorauer, representam as doenças mais comuns e destrutivas da cultura da batata no Brasil e no mundo.

Favorecida por períodos de alta umidade e temperaturas amenas, a requeima pode ocorrer em qualquer fase da cultura, afetando drasticamente folhas, hastes, pecíolos e tubérculos. A doença reduz de forma significativa a produtividade e a qualidade dos vegetais, podendo causar elevadas perdas em curto espaço de tempo.

A pinta preta é caracterizada pela redução da área foliar, queda de vigor, quebra de hastes e redução da produção. Comum em períodos quentes e úmidos, a doença geralmente está associada ao amadurecimento dos tecidos e torna-se mais agressiva a partir das fases de formação e crescimento dos tubérculos.

Assim, detectar e diagnosticar essas doenças em uma vegetação tão importante é a base desta pesquisa, para auxiliar os agricultores na identificação e classificação de doenças em plantas de batata de forma mais rápida e precisa, melhorando o rendimento da colheita, aumentando o lucro do

agricultor e contribuindo para a economia do país.

Anteriormente, muitos pesquisadores do campo da visão computacional e processamento de imagem propuseram o uso de técnicas de processamento de imagem tradicionais como LBP [4] e agrupamento K-means [5] para detectar doenças foliáceas. Modelos baseados em *Deep Learning*, devido a sua capacidade de captação de dados e reconhecimento de padrões, são mais eficazes em funções de mapeamento e, portanto, são melhores para aplicações em reconhecimento/detecção de imagens.

Desta forma, neste artigo, apresenta-se um modelo de *Deep Learning* com diversos classificadores apto a detectar doenças das folhas de batata. O artigo está dividido em seções, na qual a Seção I é uma introdução, a Seção II apresenta uma revisão da literatura, a Seção III descreve os materiais e métodos, a Seção IV expõe os resultados e discussões e na Seção V apresenta-se a conclusão.

II. REVISÃO DA LITERATURA

Existem inúmeros métodos para detectar as doenças das plantas e vários pesquisadores sugeriram diversas técnicas para diagnosticar doenças especificamente em folhas de batata. No entanto, este campo de pesquisa ainda é considerado carente de estudos [6] e se tornou uma área de pesquisa que continua a ser desenvolvida, pois a temática em questão demanda constante atualizações dos conhecimentos já produzidos, devido às modificações das doenças, ao surgimento de novas patologias e à diversificação nos modelos de produção agrícola. No que concerne à bibliografia do tema, trinta e sete artigos na área de agricultura foram publicados no período 2015-2017. Mais precisamente, foram publicados 15 artigos em 2017, 15 em 2016 e 7 em 2016. Este fato demonstra o quão nova e moderna é a utilização do Deep Learning como técnica de mapeamento agrícola [7].

P. Badar et. al. [8] usaram uma abordagem de segmentação usando *K-Means Clustering* [5] em várias características de amostras de imagens de folhas de batata, como cor, textura, área, etc. e aplicaram o algoritmo *Back Propagation Neural Network* para identificar e classificar a doença na

imagem da folha, por meio da qual obtiveram uma precisão de classificação de 92%. M. Islam et. al. [9] utilizaram o conceito de segmentação de imagem na classe de folha de batata do conjunto de dados PlantVillage [10] e, em seguida, aplicaram uma *multiclass Support Vector Machine* (SVM) nesta imagem segmentada para classificar as doenças, obtendo desta forma 95% de precisão. C. G. Li et. al. [11] apresentaram uma proposta de segmentação de imagens para a identificação e classificação de doenças fúngicas no conjunto de dados de folhas de videira. Neste trabalho, fora utilizado o agrupamento K-Means para extrair parâmetros de cor, textura e forma das imagens, e aplicara-se a técnica *Support Vector Machine* (SVM) às características extraídas para a classificação das doenças.

A técnica de *Deep Learning* [12] tem demonstrado significativo êxito em diversas atividades de percepção visual, como detecção de textos [13], [14], identificação de vítimas [15], [16], rastreamento de alvos [17], [18] e reconhecimento de objetos [19]–[21]. No trabalho citado, [15] foi provado teórica e empiricamente que as últimas camadas da rede podem capturar mais informações semânticas ou abstrações. Assim, se torna mais robusto à variação de posição, cor, escala e deformação do objeto, ou seja, pode ser adequado para classificar as doenças foliáceas de forma precisa. Recentemente, a abordagem Faster R-CNN [22] foi adotada para fins de identificação e classificação de imagens [23], [24]. A. Ramcharan et. al. [25] usou o conceito de aprendizagem de transferência para imagens de doenças da mandioca.

III. MATERIAIS E MÉTODOS

Deep Learning é uma abordagem de aprendizado de máquina eficaz, que acelerou o processo de aprendizagem de máquina tradicional da engenharia. Esta abordagem é baseada em redes neurais artificiais, ou seja, modelos matemáticos que replicam os neurônios e sinapses, reproduzindo os princípios gerais da função cerebral [26]. Atualmente, técnicas de aprendizado de máquina são utilizadas para resolver problemas nos mais variados domínios de aplicação. Dentre as aplicações mais comuns está o reconhecimento e classificação de

imagens, onde técnicas de *Deep Learning* podem apresentar ótimos resultados.

Para implementar uma rede neural artificial, uma das bibliotecas mais comuns é a biblioteca TensorFlow, pois ela fornece todas as bibliotecas relacionadas a esta estrutura. Com a ajuda do TensorFlow, é possível realizar tarefas de classificação em texto e imagens. Neste estudo, foi aplicada a arquitetura de rede neural convolucional (CNN), com o modelo VGG19 e para o treinamento deste modelo utilizou-se o Jupyter Notebook, a biblioteca TensorFlow e a API Keras do TensorFlow, que é uma API de alto nível para construir e treinar modelos de *Deep Learning*.

A. TensorFlow

O TensorFlow é uma biblioteca de código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas. É um sistema para criação e treinamento de redes neurais para detectar e compreender padrões e correlações, de forma análoga à forma como humanos aprendem e raciocinam [27]. É uma biblioteca usada tanto para a pesquisa quanto para a produção no Google, e está aos poucos substituindo seu antecessor de código proprietário, DistBelief.

O TensorFlow foi desenvolvido pela equipe Google Brain para uso interno na empresa. Foi lançado sob a licença de código aberto Apache 2.0 em 9 de novembro de 2015.

B. Keras

O Keras é uma biblioteca de rede neural de código aberto escrita em Python. É capaz de rodar a partir de TensorFlow, Microsoft Cognitive Toolkit, R, Theano, ou PlaidML [28]. Projetado para permitir uma experiência rápida com redes neurais profundas, ele se concentra em ser fácil de usar, modular e extensível. Foi desenvolvido como parte da pesquisa do projeto ONEIROS (Sistema Operacional de Robô Inteligente Neuro-Eletrônico Aberto, do inglês Open-ended Neuro-Electronic Intelligent Robot Operating System) [29].

Em 2017, a equipe do TensorFlow do Google decidiu aprovar o Keras na biblioteca principal do TensorFlow [30].

O Keras contém várias funções para construir partes importantes de redes neurais, como: camadas, funções de perda, funções de ativação, otimizadores, entre outras. O código está disponível no GitHub e os fóruns de suporte da comunidade incluem a página de problemas do GitHub e um canal do Slack. Além das redes neurais padrão, ele tem suporte para redes neurais convolucionais e recorrentes. Ele suporta outras camadas comuns, como camadas de drop-out, normalização em lote e pooling.

C. Jupyter Notebook

O Jupyter Notebook é um aplicativo web de código aberto que permite criar e compartilhar documentos que contêm código, equações, visualizações e texto narrativo. Os usos incluem: limpeza e transformação de dados, simulação numérica, modelagem estatística, visualização de dados, aprendizado de máquina e muito mais.

D. Rede Neural Convolucional (CNN)

As redes neurais convolucionais (CNNs), também conhecidas como ConNet, surgiram do estudo do córtex visual do cérebro e são usadas no reconhecimento de imagens desde a década de 1980. Devido ao aumento do poder computacional e à quantidade de dados de treinamento disponíveis, as CNNs conseguiram obter desempenho sobre-humano em algumas tarefas visuais complexas, possibilitando serviços de pesquisa de imagens, carros autônomos, sistemas automáticos de classificação de vídeo e muito mais. Além disso, as CNNs não se restringem à percepção visual: elas também são bem-sucedidas em outras tarefas, como reconhecimento de voz ou processamento de linguagem natural (PLN).

A arquitetura do modelo CNN é composta por camadas, sendo elas: camada convolucional, camada de unidade linear retificada (ReLU), camada de conjugação (*pooling*), camada de desconsideração de neurônios (*drop out*) e camadas totalmente conectadas (*fully connected*).

O bloco de construção mais importante de uma CNN é a camada convolucional: identificam as características pela aplicação de um conjunto de

filtros que extraem informações úteis de uma determinada imagem. Estas camadas detectam padrões na imagem de entrada, e após a aplicação de filtros, os dados são passados na camada ReLU. Um exemplo ilustrativo de uma camada convolucional é apresentado na Fig. 1.

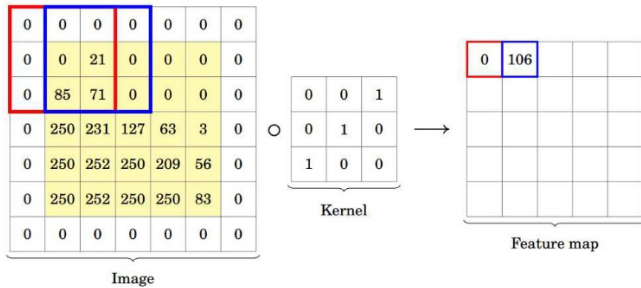


Fig. 1. Camada convolucional, utilizando um filtro 3x3. Fonte: [31]

A camada ReLU define todos os valores negativos da matriz convolucionada para zero usando a função $\max(a, 0)$, desta forma, os dados processados são inseridos na camada de *pooling*. Um exemplo genérico de camada convolucional completa é mostrado na Fig. 2.

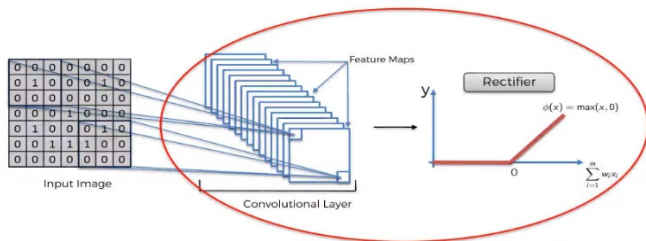


Fig. 2. Exemplo genérico de camada convolucional completa. Fonte: [32]

Posteriormente, a camada *pooling* reduz o tamanho das entradas e acelera o processamento. Os diferentes parâmetros na camada de *pooling* do modelo CNN são: o tamanho do filtro, a passada (*stride*) e o *pooling* máximo ou médio. Cabe mencionar que um modelo CNN pode conter múltiplas camadas convolucionais e de *pooling*. Com relação às camadas totalmente conectadas, elas estão presentes ao final de um modelo CNN, ou seja, cada nó está conectado a todos os nós da camada anterior em camadas totalmente conectadas - camadas totalmente conectadas são

camadas fortemente orientadas por dados do modelo CNN. Uma camada totalmente conectada realiza a tarefa de classificação da imagem em diferentes classes na camada de saída.

E. Conjunto de Dados

Kaggle é um repositório de código aberto que fornece a base de dados PlantVillage Dataset [10] para fins de pesquisa. O conjunto de dados compreende aproximadamente 55.000 imagens rotuladas de folhas saudáveis e folhas infectadas de várias frutas e vegetais como maçã, mirtilo, cereja, uva, pêssigo, pimenta, laranja, tomate, batata, etc. As pastas de frutas e vegetais tem dois tipos de imagens, ou seja, coloridas e em tons de cinza. As diferentes culturas contêm mais de um tipo de doença foliácea e, para classificação, qualquer tipo é considerado uma classe separada de doença. O conjunto de dados [10] foi dividido em dois grupos: um deles composto pelas imagens de folhas com fundo e o outro sem o fundo das imagens. Um exemplo de amostras de imagens para cada classe é apresentado na Fig. 3.

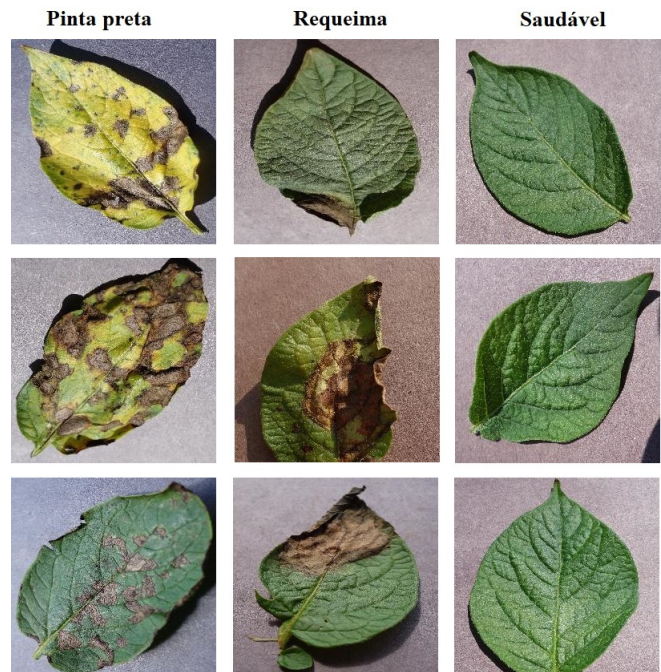


Fig. 3. Amostras de imagens para cada classe. Fonte: [10]

O número de imagens em uma determinada classe não é uniforme, varia de 152 imagens a 1000 imagens. Neste artigo, foram utilizadas apenas

imagens de folhas de batata para classificação, compondo três classes: imagens de folhas não-saudáveis com pinta preta, imagens de folhas não-saudáveis com requeima e imagens de folhas saudáveis.

O conjunto de dados foi dividido em duas partes, uma para treinamento e outra para teste. A divisão do conjunto de dados é de proporção 80/20 aleatoriamente, ou seja, 80% para o conjunto de dados de treinamento e 20% restantes para o conjunto de dados de teste. A divisão dos dados de treino-teste são mostrados na Tabela I.

Rótulo	Categoria	Número de imagens	Amostras de treinamento	Amostras de teste
0	Pinta preta	1000	800	200
1	Requeima	1000	800	200
2	Saudável	152	121	31

TABLE I

NÚMERO DE AMOSTRAS, CONJUNTO DE TREINAMENTO E CONJUNTO DE TESTE DO MODELO PROPOSTO.

F. Modelo Proposto

A metodologia proposta neste artigo incluiu as seguintes etapas principais: aquisição de dados, pré-processamento de dados, aumento de dados e classificação de imagens, como é apresentado no diagrama de blocos através da Fig. 4.

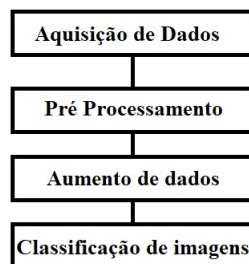


Fig. 4. Diagrama de bloco do modelo proposto.

Primeiramente, foi aplicado um pré-processamento no conjunto de dados, a fim de minimizar o ruído na imagem, cortando a parte da imagem que não é a região de interesse. Se houver ruído excessivo na imagem, ela não será usada. Imagens coletadas de fontes múltiplas de tamanhos diferentes devem ser redimensionadas

para 256x256 pixels para padronizar as imagens de entrada no conjunto de dados. Em seguida, foi aplicada a técnica de aumento de dados, que é uma técnica de manipulação de dados sem perder as principais características destes dados, para ampliar o tamanho do conjunto de dados e diminuir a generalidade do modelo, a fim de obter um melhor desempenho.

Posteriormente, foi utilizado o modelo de estrutura VGGNet, que tem sua macro arquitetura apresentada na Fig. 5, no qual o dado de entrada passa por 5 blocos de camada convolucional, em que cada bloco consiste em um aumento no número de filtros 3x3. A camada de ativação ReLU foi aplicada em cada bloco para reconhecer a não linearidade para que o modelo possa se adaptar mais facilmente a vários dados. As técnicas de *drop out* são usadas com uma probabilidade fixa de 0,25 para reduzir as condições de sobreajuste. A camada de *pooling* máximo separa os blocos. Três camadas totalmente conectadas seguem cinco blocos de camada convolucional. A camada *softmax* é a última camada a produzir probabilidades de classe. A arquitetura VGG19 é representada na Fig. 6.

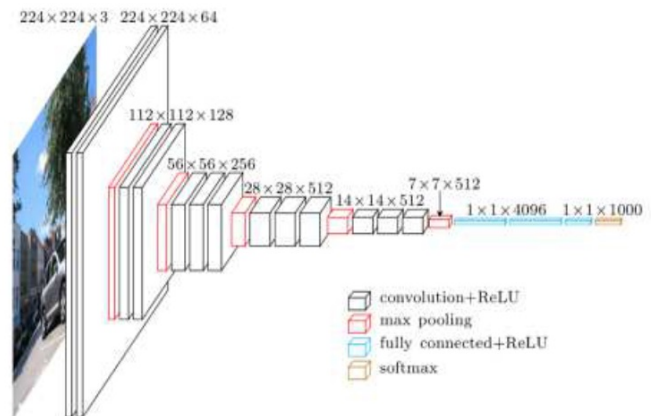


Fig. 5. Arquitetura macro de uma VGGNet. Fonte: [33]

Por fim, o modelo CNN foi treinado com o conjunto de dados PlantVillage.

IV. RESULTADOS E DISCUSSÕES

O treinamento do conjunto de dados foi realizado utilizando o método de redes neurais

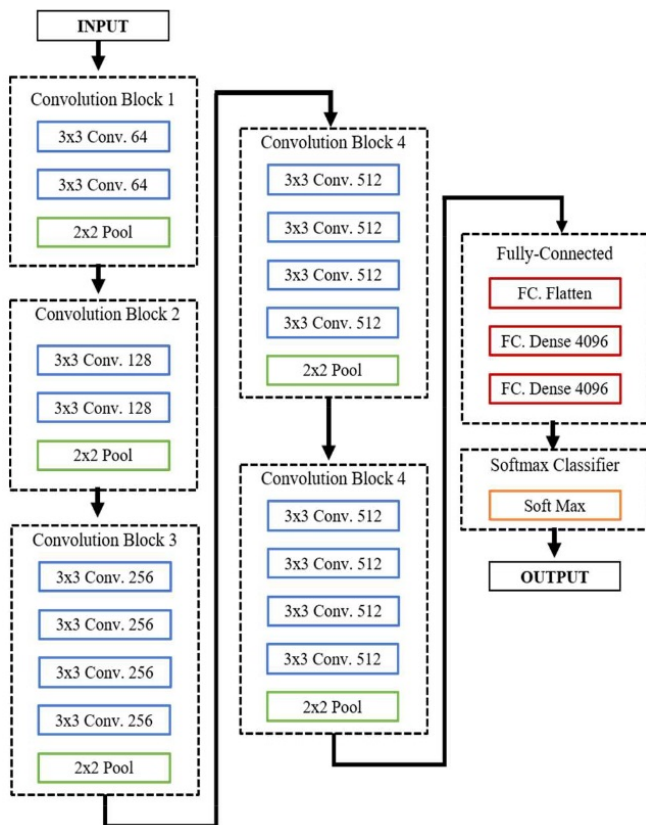


Fig. 6. Arquitetura da rede VGG19. Fonte: [34]

convolucionais usando o modelo de arquitetura da família VGGNet, especificamente VGG19, que possui 19 camadas. Foram especificadas neste estudo 250 épocas com tamanho de lote de 64 e uma taxa de aprendizado de 0,01 para melhorar o desempenho do modelo. O processo de aprendizagem significa que o algoritmo com o método proposto busca encontrar valores no conjunto de dados de imagens para ser capaz de reconhecer novas imagens. Cada iteração da época deve corresponder ao valor da imagem que está sendo treinada. Os resultados das épocas serão registrados para determinar o valor de perda e precisão. A perda é o indicativo de um valor ruim no modelo, o valor da perda obtido deve ser próximo ou igual a zero e o valor de precisão é o parâmetro que define o nível de sucesso do sistema na classificação de objetos. Os resultados dos valores de perda e precisão são apresentados nos gráficos do progresso de treinamento apresentados na Fig. 7, obtendo valores de perdas próximos de zero e precisão após a 250ª época chega a 90%.

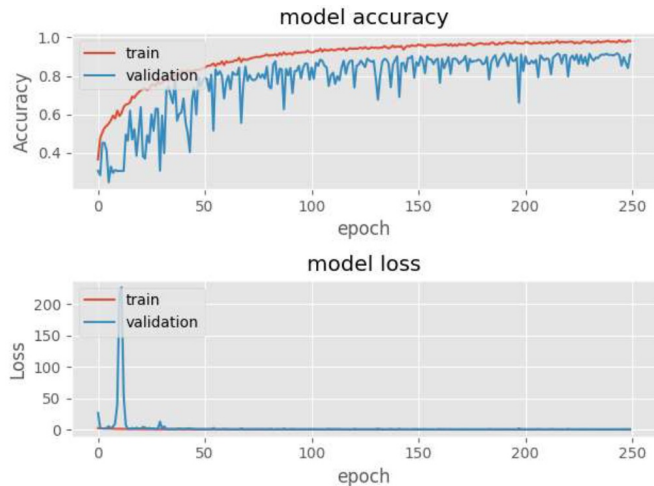


Fig. 7. Gráfico do Processo de Treinamento.

O conjunto de treinamento foi composto por 1721 imagens: 800 imagens para a classe não-saudável pinta preta, 800 imagens para a classe não-saudável requeima e 121 imagens para a classe saudável.

No que diz respeito ao conjunto usado para teste, este foi composto de 431 imagens: 200 imagens para a classe não-saudável pinta preta, 200 imagens para a classe não-saudável requeima e 31 imagens para a classe saudável.

Uma matriz de confusão foi utilizada para analisar informações sobre o desempenho do conjunto de dados de teste no modelo treinado. A Tabela II mostra a matriz de confusão gerada, a qual foi útil para o cálculo das medidas de desempenho como acurácia, precisão, sensibilidade e especificidade do modelo [35].

Matriz de Confusão				
		Classe de previsão		
		0	1	2
Classe real	0	188	12	0
	1	0	182	18
	2	0	2	29

TABLE II
MATRIZ DE CONFUSÃO.

O total de imagens de teste utilizado para qualquer classe é obtido pela soma de todos os itens das linhas correspondentes a classe, ou seja, os

valores verdadeiros positivos (VP) e os valores falsos negativos (FN), $(VP + FN)$. Por exemplo, para a pinta-preta, linha correspondente ao rótulo 0, o número total de imagens de teste foi: $188 + 12 + 0 = 200$.

Do mesmo modo, o número de falsos positivos (FP) para uma classe é a soma dos valores na coluna correspondente, excluindo os verdadeiros positivos para essa classe.

O número total de verdadeiros negativos (VN) para uma determinada classe será a soma de todos os valores das colunas e linhas, exceto a coluna e linha daquela classe.

A precisão geral do modelo é definida pela soma dos verdadeiros positivos de todas as classes (soma de todos os elementos diagonais), dividido pelo número total utilizado para as classificações (soma de todos os itens da matriz de confusão), assim a precisão do modelo CNN proposto foi: $\text{Precisão} = (188+182+29)/431 = 399/431 = 0,9257 (92,57\%)$. A precisão de uma classe específica é dada por $VP/(VP+FP)$. Por exemplo, a precisão do modelo CNN de previsão de folhas saudáveis (coluna 2) folhas foi de $29/(29+18) = 29/47 = 0,617 (61,7\%)$.

V. CONCLUSÃO

No presente estudo, foi analisada a viabilidade da aplicação de um modelo CNN para a identificação de doenças de plantas de batata baseada em imagens de suas folhas. Utilizou-se um conjunto de dados contendo 2152 imagens de folhas de batata divididas em três classes. O modelo CNN teve um bom desempenho no conjunto de dados de teste, com uma precisão geral de 92,57%. Ele foi treinado com 80% das imagens e testado com 20% delas. Portanto, conclui-se, com base na precisão geral, que o modelo CNN utilizado é adequado para a detecção e diagnóstico de doenças nas plantas de batata. Para além, este sistema também pode ser integrado a mini-drones, a fim de detectar em tempo real doenças nas plantas ao longo de grandes áreas cultivadas.

Embora este modelo CNN tenha sido treinado com o conjunto de dados da PlantVillage com apenas 3 classes, ele pôde identificar se a planta tinha uma doença ou não. Além disso, mais imagens do ambiente real podem ser adicionadas ao

conjunto de dados de treinamento para aperfeiçoar a precisão na identificação de imagens de folhas em condições reais.

AGRADECIMENTOS

Agradecemos à Universidade Federal de São Carlos (UFSCar) pela disponibilidade de acesso ao ambiente CAFE, que possibilitou encontrar recursos bibliográficos necessários para a realização desta pesquisa.

Eu, Diego Ramos Biondo, também agradeço ao professor orientador Roberto Santos Inoue, pela paciência e apoio, que foram de suma importância para a realização desta pesquisa.

REFERENCES

- [1] A. da Silva Pereira and F. Akiyoshi Suinaga, "Sistema de produção da batata - introdução e importância econômica," July 2016. [Online]. Available: https://www.spo.cnptia.embrapa.br/conteudo?p_p_id=conteudoportlet_WAR_sistemasdeproducao16_lga1ceportletp_p_lifecycle=0p_p_state=normalp_p_mode=viewp_p_col_id=column-2p_p_col_count=1p_r.p.-76293187_sistemaProducaoId=8803p_r.p.-996514994_topicoId=1301
- [2] R. Arora and S. Sharma, *Pre and Post Harvest Diseases of Potato and Their Management*, 07 2014, pp. 149–183.
- [3] "List of potato diseases," 2020. [Online]. Available: https://en.wikipedia.org/wiki/List_of_potato_diseases
- [4] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 765–781, 2011.
- [5] Y. Li and H. Wu, "A clustering method based on k-means algorithm," *Physics Procedia*, vol. 25, pp. 1104 – 1109, 2012, international Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1875389212006220>
- [6] M. Van Hulle, S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Hindawi Publishing Corporation*, 2016, computational Intelligence and Neuroscience. [Online]. Available: <https://www.hindawi.com/journals/cin/2016/3289801/>
- [7] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70 – 90, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169917308803>
- [8] . G. Athanikar and M. P. Badar, "Potato leaf diseases detection and classification system mr," 2016.
- [9] M. Islam, Anh Dinh, K. Wahid, and P. Bhowmik, "Detection of potato diseases using image segmentation and multiclass support vector machine," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, pp. 1–4.

- [10] A. Ali, "Plantvillage dataset," September 2019. [Online]. Available: <https://www.kaggle.com/xabdallahali/plantvillage-dataset>
- [11] G. Li, Z. Ma, and H. Wang, "Image recognition of grape downy mildew and grape powdery mildew based on support vector machine," vol. 370, no. 10 2011, pp. 151–162.
- [12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.
- [13] A. Risnumawan, I. A. Sulistijono, J. Abawajy, and Y. Saadi, "Text detection in low resolution scene images using convolutional neural network," 08 2016.
- [14] M. L. Afakh, A. Risnumawan, M. E. Anggraeni, M. N. Tamara, and E. S. Ningrum, "Aksara jawa text detection in scene images using convolutional neural network," in *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 2017, pp. 77–82.
- [15] I. A. Sulistijono and A. Risnumawan, "From concrete to abstract: Multilayer neural networks for disaster victims detection," in *2016 International Electronics Symposium (IES)*, 2016, pp. 93–98.
- [16] I. A. Sulistijono, T. Imansyah, M. Muhajir, E. Sutoyo, M. K. Anwar, E. Satriyanto, A. Basuki, and A. Risnumawan, "Implementation of victims detection framework on post disaster scenario," in *2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, 2018, pp. 253–259.
- [17] M. K. Anwar, A. Risnumawan, A. Darmawan, M. N. Tamara, and D. S. Purnomo, "Deep multilayer network for automatic targeting system of gun turret," in *2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, 2017, pp. 134–139.
- [18] M. K. Anwar, M. Muhajir, E. Sutoyo, M. L. Afakh, A. Risnumawan, D. S. Purnomo, E. Suryawati Ningrum, Z. Darojah, A. Darmawan, and M. N. Tamara, "Deep features representation for automatic targeting system of gun turret," in *2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, 2018, pp. 107–112.
- [19] H. Imaduddin, M. K. Anwar, M. I. Perdana, I. A. Sulistijono, and A. Risnumawan, "Indonesian vehicle license plate number detection using deep convolutional neural network," in *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 2018, pp. 158–163.
- [20] D. M. Dinama, Q. A'yun, A. D. Syahroni, I. Adji Sulistijono, and A. Risnumawan, "Human detection and tracking on surveillance video footage using convolutional neural networks," in *2019 International Electronics Symposium (IES)*, 2019, pp. 534–538.
- [21] A. Risnumawan, M. I. Perdana, A. H. Hidayatulloh, A. K. Rizal, I. A. Sulistijono, A. Basuki, and R. Febrianto, "Automatic detection of wrecked airplanes from uav images," *EMITTER International Journal of Engineering Technology*, vol. 7, no. 2, pp. 570–585, December 2019. [Online]. Available: <https://emitter.pens.ac.id/index.php/emitter/article/view/424>
- [22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [23] D. Oppenheim, G. Shani, O. Erlich, and L. Tsrer, "Using deep learning for image-based potato tuber disease detection," *Phytopathology*, vol. 109, no. 6, pp. 1083–1087, 2019, pMID: 30543489. [Online]. Available: <https://doi.org/10.1094/PHYTO-08-18-0288-R>
- [24] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, p. 2022, 09 2017.
- [25] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. Hughes, "Using transfer learning for image-based cassava disease detection," *Frontiers in Plant Science*, vol. 8, 06 2017.
- [26] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311 – 318, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169917311742>
- [27] "Uma plataforma completa de código aberto para machine learning." [Online]. Available: <https://www.tensorflow.org>
- [28] "Keras backends." [Online]. Available: <https://keras.io/about/>
- [29] "Keras documentatation." [Online]. Available: <https://keras.io/why-this-name-keras>
- [30] "Module: tf.keras." [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras
- [31] V. PAVLOVSKY, "Introduction to convolutional neural networks," 2017. [Online]. Available: <https://www.vaetas.cz/blog/intro-convolutional-neural-networks>
- [32] "Superdatascience," 2017. [Online]. Available: <https://www.superdatascience.com/deep-learning>
- [33] D. FROSSARD, "Vgg in tensorflow," 2016. [Online]. Available: <https://www.cs.toronto.edu/~frossard/post/vgg16/>
- [34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.
- [35] A. Singh, S. Walia, and S. Kumar, "P3pga: Multi-population 3 parent genetic algorithm and its application to routing in wmn," *International Journal of Advanced Research in Computer Science*, vol. 8, pp. 968–975, 2017.

¹ Diego Ramos Biondo, discente no curso de graduação em Engenharia Elétrica, no Departamento de Engenharia Elétrica (DEE) da Universidade Federal de São Carlos (UFSCar), São Paulo, Brasil diegoramosbiondo@estudante.ufscar.br

² Roberto Santos Inoue, doutor em Engenharia Elétrica pela Universidade de São Paulo (EESC/USP 2012). Atualmente é Professor Adjunto III pela Universidade Federal de São Carlos, no Departamento de Computação (DC) da Universidade Federal de São Carlos (UFSCar), São Paulo, Brasil rsinoue@ufscar.br