

Modelo para Classificação Fitopatológica em Vegetais *Phaseolus vulgaris*

Igor Gonçalves de Souza
Mestrado em Telecomunicações
Instituto Nacional de Telecomunicações
Santa Rita do Sapucaí, Brasil
igor.souza@mtel.inatel.br

Ítalo Augusto de Sousa Tacca
Mestrado em Telecomunicações
Instituto Nacional de Telecomunicações
Santa Rita do Sapucaí, Brasil
italo.tacca@mtel.inatel.br

Abstract—In this article, the process of developing a Convolutional Neural Network model for classifying diseases in *Phaseolus Vulgaris* plants, also known as beans, is presented. The functioning of this specific type of neural network is detailed, describing the definition and training of the classification model. Subsequently, the results obtained in this process are presented, and an application for the proposed model is discussed.

Index Terms—Beans, Disease, Machine Learning.

Resumo—Neste artigo, apresenta-se o processo de desenvolvimento de um modelo de Rede Neural Convolucional destinado à classificação de doenças em plantas da espécie *Phaseolus Vulgaris*, comumente conhecidas como feijão. Explora-se minuciosamente o funcionamento dessa categoria particular de rede neural, abordando a definição e o procedimento de treinamento do modelo de classificação. Em seguida, são expostos os resultados obtidos nesse processo, seguidos de uma discussão sobre uma possível aplicação para o modelo proposto.

Palavras Chave—Aprendizado de Máquina, Doença, Feijão.

I. INTRODUÇÃO

Atualmente, a agricultura configura-se como uma das principais atividades econômicas no Brasil. A produção de grãos e cereais, por exemplo, representa 35% das exportações do país, sendo soja, milho e algodão os principais produtos para este tipo de safra. Refletindo esses números, estima-se que, em 2022, exportou-se mais de 100 milhões de toneladas de soja e derivados, o que correspondeu a uma movimentação de, aproximadamente, 60 bilhões de dólares [1].

Pragas e outras doenças na agricultura podem resultar em diminuição no volume de produção, em prejuízos à qualidade dos produtos, além de provocar a morte de plantas e até destruir cultivos inteiros. Um estudo realizado pelo Centro de Estudos Avançados em Economia Aplicada - CEPEA, analisou os impactos econômicos gerados pelas principais pragas que atacam as plantações de soja, milho e algodão no Brasil. Apenas para o controle da Ferrugem Asiática, principal doença da cultura de soja, os produtores investem diretamente R\$ 5,75 bilhões anualmente. Porém, sem o controle adequado, a doença alcançaria patamares alarmantes, chegando a uma redução de até 30% na produtividade [2].

Prejuízos agrícolas causados pelo não controle de pragas e doenças têm impactos significativos nos aumentos dos preços dos produtos, penalizando a sociedade como um todo com elevadas taxas de inflação alimentar. A Figura 1 ilustra os

efeitos econômicos no setor causados pelas principais doenças na cultura da soja.



Figure 1. Impactos econômicos do não controle de pragas na soja [2].

Apesar disso, o Brasil ainda demonstra um baixo grau de adoção de tecnologia no campo para otimizar sua capacidade de produção. Segundo dados do censo agropecuário do Instituto Brasileiro de Geografia e Estatística - IBGE, a tecnologia foi a responsável por um aumento de quase 20% no crescimento da produção de grãos entre 1996 e 2006, de 50% para 70% [3]. Concomitantemente à esse aumento, segundo o relatório “A Dinâmica do Crescimento das Exportações do Agronegócio Brasileiro” do Ipea, o Brasil assumiu, em 2013, a liderança no *ranking* mundial de produção e exportação agrícola para a soja, o que evidencia a importância do avanço tecnológico nesse cenário de aplicação [4].

Considerando os desafios enfrentados pela agricultura brasileira, marcados pelos impactos econômicos causados por pragas e doenças nas plantações, bem como a limitada adoção de tecnologia no setor para mitigar esse problema, o objetivo deste trabalho é propor um modelo de aprendizado de máquina baseado em visão computacional para classificação das folhas das plantas, distinguindo entre aquelas saudáveis e afetadas por doenças. A implementação do modelo busca não apenas atenuar os prejuízos econômicos decorrentes dos males nas lavouras, mas também promover uma abordagem tecnológica para o aprimoramento da produtividade agrícola no Brasil.

O trabalho está dividido em 7 capítulos, em que, além da Introdução, o capítulo II comenta sobre trabalhos relacionados

ao tema; o capítulo III descreve os métodos de proposição do modelo; o capítulo IV apresenta as etapas de treinamento e como o modelo da rede neural foi concebido; o capítulo V indica os resultados obtidos após a etapa de testes; o capítulo VI exemplifica uma aplicação prática; e o capítulo VII expõe as conclusões sobre o modelo.

II. TRABALHOS RELACIONADOS

Diversos trabalhos têm sido desenvolvidos nos últimos anos para a classificação de doenças em plantas, especialmente em vegetais que fazem parte da cadeia alimentar humana, uma vez que, com o crescimento da população mundial, busca-se cada vez mais a implementação de técnicas que possibilitem o aumento da produtividade das safras e uma menor utilização de agrotóxicos.

Nesse contexto, o treinamento de modelos de redes neurais convolucionais, assim como a utilização de técnicas de *transfer learning* tem se mostrado altamente eficazes para a classificação de doenças em diferentes espécies de plantas. Chen et al. em [5] demonstraram um modelo capaz de detectar doenças que atingem folhas de chá com acurácia de 90,16%, utilizando uma rede neural convolucional treinada do zero.

Liu et al. em [6] trabalharam com um modelo de rede neural convolucional, também treinada do zero, para a detecção de doenças que atingem as folhas de macieiras. A acurácia alcançada foi de 97,62%. Já Liang et al. em [7] apresentaram um estudo comparativo entre o desempenho de redes neurais convolucionais tradicional e com SVM (*Support Vector Machine*) para a classificação de doenças no arroz, em que concluíram que ambos os modelos apresentam desempenho muito próximo, com uma acurácia de 95,83% do primeiro, contra 95,82% do segundo.

Zaki et al. em [8] apresentaram um modelo de rede MobileNetV2 com acurácia de 90% desenvolvido a partir *transfer learning* para a classificação de doenças do tomate. Vinutha et al. em [9] também trabalharam com a MobileNetV2, no entanto treinaram-na para a classificação de 15 doenças em 8 espécies de vegetais, alcançando uma acurácia de 99%. E. Elfatimi et al. em [10] implementaram um modelo baseado na MobileNetV2 para classificação de doenças que atingem as folhas do feijão, com acurácia de 92,97%.

A existência de diversos trabalhos baseados na técnica de *transfer learning* utilizando a rede MobileNetV2 deve-se à eficiência computacional desse modelo, que viabiliza implementações práticas em dispositivos embarcados como *smartphones* ou *drones*. Além disso, os trabalhos citados nesta seção utilizaram o Tensorflow, uma biblioteca de código aberto para o desenvolvimento e o treinamento dos modelos de aprendizado de máquina.

Este trabalho, portanto, busca apresentar o desenvolvimento e o desempenho de um modelo para a classificação do estado de saúde das folhas da espécie *Phaseolus Vulgaris*, utilizando a técnica de *transfer learning* com a rede MobileNetV2 e a plataforma Edge Impulse.

III. METODOLOGIA

Nesta seção, serão apresentados o *dataset* e o modelo de rede neural utilizados além de características da plataforma de desenvolvimento aplicados à este trabalho.

A. Dataset Beans

O conjunto de dados *Beans* consiste em 1.167 imagens com tamanho 160x160 pixels de feijões tiradas no campo usando câmeras de *smartphones*, composto por 3 classes: 2 de doenças e a classe saudável (*Healthy*). As doenças representadas incluem Mancha Angular da Folha (*Angular Leaf Spot*) - definida, apenas, como “Manchada”, e Ferrugem do Feijão (*Bean Rust*) [11].

As Figuras 2 e 3 ilustram duas imagens pertencentes ao conjunto de dados, representando, respectivamente, folhas Manchada e com Ferrugem do Feijão.



Figure 2. Manchada [11].



Figure 3. Ferrugem [11].

B. Edge Impulse

O Edge Impulse é uma plataforma de desenvolvimento para sistemas de aprendizado de máquina embarcado (*Edge Machine Learning*) projetado para facilitar o processo de coleta, treinamento e implementação de modelos de *Machine Learning* em dispositivos como microcontroladores, módulos IoT e outros dispositivos de *hardware* integrado [12].

A plataforma oferece várias funcionalidades para implementar soluções de aprendizado de máquina:

- **aquisição de dados:** ferramentas para adquirir dados de sensores em dispositivos como câmera, acelerômetro e microfone para treinamento;
- **treinamento do modelo:** interface para treinamento de modelos de aprendizado de máquina com os dados coletados. Os modelos podem ser treinados para realizar tarefas específicas, como reconhecimento de padrões, classificação ou detecção de eventos;
- **implementação em dispositivos de borda:** modelo implementado diretamente em dispositivos de borda, permitindo que os modelos sejam executados localmente no dispositivo, sem depender de uma conexão com a nuvem para inferência. Também há integrações com plataformas como Arduino, Raspberry Pi, TensorFlow Lite;
- **monitoramento:** desempenho e classificação dos modelos em tempo real;

- **otimização de parâmetros:** algoritmo de otimização que ajuda a encontrar o melhor modelo de aprendizado de máquina para a aplicação.

C. EON Tuner e Transfer Learning

O *EON Tuner* é uma otimização hiper paramétrica disponível no Edge Impulse que seleciona o melhor modelo de aprendizado de máquina incorporado para a aplicação. O algoritmo analisa os dados de entrada, possíveis blocos de processamento de sinal e arquiteturas de redes neurais e fornece uma visão geral de possíveis arquiteturas de modelos que atenderão aos requisitos [13].

O processamento do *EON Tuner* utiliza, entre outras descrições, *Transfer Learning*, uma técnica em aprendizado de máquina que envolve o uso de modelos pré-treinados em tarefas específicas e a adaptação desses modelos para novas tarefas relacionadas. Em vez de treinar um modelo do zero, aproveita-se o conhecimento adquirido por um modelo em uma tarefa anterior e o aplica a uma tarefa relacionada.

A ideia é que modelos treinados em grandes conjuntos de dados aprendam representações úteis e generalizáveis dos dados e sejam aplicadas à tarefas específicas, permitindo uma inicialização mais eficiente e melhor desempenho com conjuntos de dados menores.

São duas abordagens principais para o *Transfer Learning*:

- **ajuste fino:** um modelo pré-treinado é utilizado como ponto de partida e suas camadas são ajustadas para se adequarem ao conjunto de dados e à nova tarefa. Isso envolve treinar o modelo por algumas iterações com o novo conjunto de dados, ajustando pesos das camadas específicas.
- **extração de características:** as camadas do modelo são mantidas fixas, e apenas a última camada é substituída e treinada para a nova tarefa, especialmente útil quando se tem um conjunto de dados pequeno para evitar o *overfitting*.

D. Dropout

Dropout é uma técnica de regularização em redes neurais para combater o *overfitting*, ou seja, evitar que o modelo se ajuste demais aos dados de treinamento e, conseqüentemente, tenha um desempenho inferior em novos dados.

O processo descarta, aleatoriamente, durante o treinamento um conjunto de neurônios em uma camada da rede neural, definindo suas saídas como zero. Essa operação é realizada de forma estocástica para cada exemplo e época de treinamento. Dessa forma, a rede é treinada de maneira robusta, para não depender fortemente de nenhum neurônio específico.

A camada de *dropout*, geralmente, é aplicada após as camadas de ativação em redes neurais profundas, sendo eficaz na melhoria do desempenho de redes neurais para classificação de imagens, em que o número de parâmetros é grande e o risco de sobre ajuste é significativo.

E. Redes Neurais Convolucionais - CNNs

As redes neurais convolucionais são um tipo de rede neural projetado para processar dados em grade, especialmente em tarefas de visão computacional, como reconhecimento de imagem e detecção de objetos. A rede é formada por:

- **camadas convolucionais:** filtros que capturam padrões específicos, como bordas, texturas ou formas;
- **camadas de *pooling*:** reduzem a resolução da entrada, mantendo as características mais importantes, a fim de reduzir o processamento computacional;
- **camadas densas:** transformam as características aprendidas nas camadas anteriores em uma saída final;
- **funções de ativação:** são aplicadas para introduzir não linearidade na rede, ajudando a capturar relações mais complexas nos dados;

Para problemas de classificação multi-classe, a função de ativação *softmax* é aplicada à saída da última camada de uma rede neural para os *logits* (saídas não normalizadas). A função opera em um vetor de entrada e produz um vetor de saída com as mesmas dimensões, em que cada elemento na saída representa a probabilidade associada à classe correspondente. A classe com a probabilidade mais alta é, geralmente, escolhida como a predição final do modelo.

F. Modelos MobileNet e MobileNetV2

Os modelos MobileNet são redes neurais convolucionais eficientes para aplicações de visão computacional. Essas arquiteturas suportam qualquer tamanho de entrada maior do que 32x32 - oferecendo melhor desempenho com tamanhos de imagem maiores, e retornam um modelo de classificação de imagem Keras, opcionalmente carregado com pesos pré-treinados no ImageNet [14].

Em particular, o modelo MobileNetV2 é muito semelhante ao MobileNet original, exceto pelo fato de utilizar blocos residuais invertidos com características de gargalo e possuir um valor drasticamente mais baixo para contagem de parâmetros. A Figura 4 detalha o funcionamento do modelo.

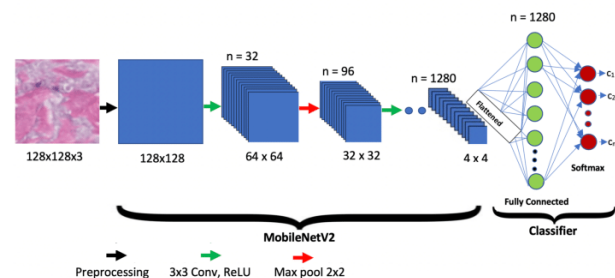


Figure 4. Estrutura do modelo MobileNetV2 [15].

A camada de expansão intermediária usa convoluções leves em profundidade para filtrar características como uma fonte de não-linearidade. Como um todo, a arquitetura do MobileNetV2 contém a camada inicial de convolução total com 32 filtros, seguida por 19 camadas de gargalo residual [15].

IV. MODELO DE APRENDIZADO DE MÁQUINA

Nesta seção, serão apresentadas as características da rede neural e as especificações para treinamento do modelo.

A. Features

No explorador de recursos do Edge Impulse, os dados do conjunto são visualizados em um gráfico em que os eixos são a saída do processamento de sinal e podem permitir validar se os dados se separam bem. Isso ajuda a encontrar dados semelhantes durante o teste do modelo e analisar possíveis erros de classificação. O Gráfico da Figura 5 mostra como o conjunto de dados está separado.

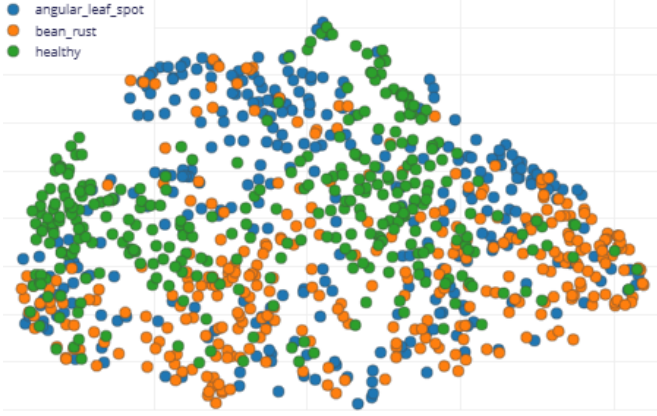


Figure 5. Feature Explorer para o dataset Beans.

Observa-se uma forte correlação entre os dados, ou seja, há ocorrência de sobreposição das classes, de forma que não é definida uma perfeita superfície de separação entre elas. Por conseguinte, é esperado que o modelo prediga determinadas imagens de entrada de forma equivocada.

B. Modelo de Rede Neural

Com o auxílio do *EON Tuner* do Edge Impulse, alguns modelos pré treinados de rede neural para visão computacional foram verificados para o conjunto de dados *Beans* com um tempo de inferência de 1000ms, sendo classificados de acordo com a acurácia. A Tabela I apresenta os resultados obtidos para os três melhores modelos.

Table I
ACURÁCIA PARA OS MODELOS DE TRANSFER LEARNING

Modelo	Acurácia
RGB - MobileNetV2 - 686	87%
GrayScale - MobileNetV2 - 8a4	82%
RGB - MobileNetV2 - 9b9	76%

Dada a maior acurácia, o modelo selecionado foi RGB - MobileNetV2 - 686. Pelas características das imagens, a Equação 1 apresenta a dimensão da camada de entrada da rede, considerando as dimensões para cada tom de cor do padrão RGB.

$$(160 \times 160) \text{ pixels} \times 3 \text{ dimensões} = 76.800 \text{ features} \quad (1)$$

Dados um problema de classificação e as três classes do conjunto de dados, a camada de saída da rede possui três neurônios com função de ativação *softmax*, indicando, entre 0 e 1, a probabilidade de o dado de entrada pertencer a cada uma das classes. O ajuste fino do modelo MobileNetV2 altera:

- para 64 o número de neurônios na camada final, anterior à camada de saída da rede;
- *dropout* definido para 0.5, ou seja, durante uma época de treinamento há uma probabilidade de 50% de que um neurônio seja temporariamente desligado.

C. Configurações de Treinamento

Conforme a Figura 7, para treinamento do modelo, o conjunto foi dividido em 80% dados de treinamento e 20%, de teste, ou seja, 933 e 234 amostras para cada etapa, respectivamente. Apresentado no Gráfico da Figura 6, o conjunto de dados de treinamento é balanceado, com divisão igualitária entre as classes.

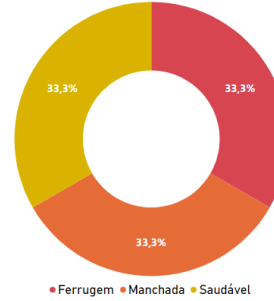


Figure 6. Distribuição das classes para treinamento

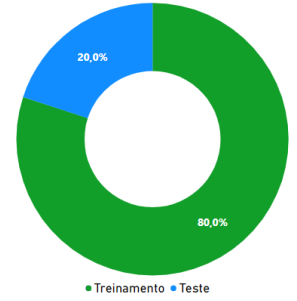


Figure 7. Proporção dos dados de treinamento e teste.

O modelo proposto foi treinado por 10 épocas com o conjunto de treinamento obtido a partir do *dataset Beans*. Definiu-se a taxa de aprendizagem como 0,0005 e o fator proporcional para o número de filtros em cada camada $\alpha = 0.35$. Além disso, o tamanho de *batch* foi definido em 32 e utilizou-se a opção de *Data Augmentation* da plataforma Edge Impulse para ampliar o conjunto de treinamento e, consequentemente, aumentar a acurácia do modelo. Os resultados obtidos são apresentados na seção V.

V. RESULTADOS

Nesta seção, serão apresentados os resultados obtidos para os conjuntos de treinamento e teste além da matriz de confusão e outras métricas acerca do modelo.

A. Acurácia e Perda

A Tabela II apresenta os resultados após as 10 épocas de treinamento do modelo. São considerados os modelos quantizados, ou seja, representação dos dados por símbolos int de 8 *bits*, e não quantizados, com símbolos *float* de 32 *bits*. Como o modelo quantizado é uma aproximação dos valores originais do modelo não quantizado, há uma diminuição da acurácia e aumento na perda para o conjunto de teste.

Table II
RESULTADOS DO TREINAMENTO DO MODELO

	Treinamento	Teste (32 bits)	Teste (8 bits)
Acurácia	91,42%	88,8%	87,2%
Perda	0,24	0,36	0,40

Percebe-se que a acurácia durante o treinamento foi a mais alta, ao passo que a perda nessa etapa é a mais baixa. Essa dinâmica é previsível, já que os dados de teste não foram expostos ao modelo durante o treinamento, resultando em um desempenho inferior quando avaliado com o conjunto de testes.

Os Gráficos da Figura 8 ilustram os valores de acurácia e perda para os conjuntos de treinamento e teste ao longo das épocas de treinamento.

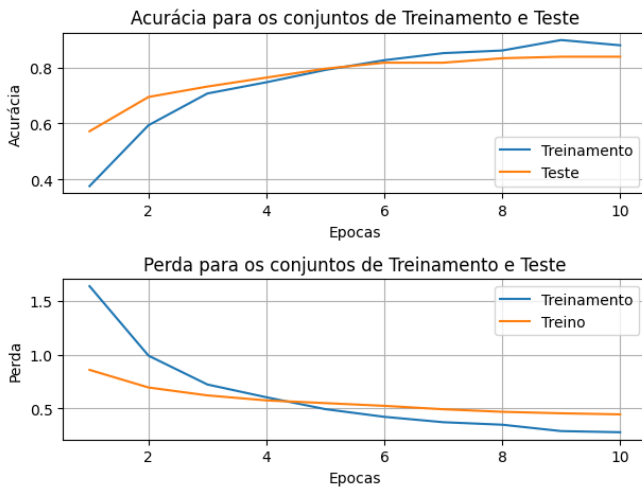


Figure 8. Acurácia e Perda para os conjuntos de Treinamento e Teste

A acurácia para o conjunto de teste se aproxima em relação ao conjunto de treinamento, ou seja, não houve sobre ajuste durante o treinamento. Ademais, a perda ao longo das épocas para os dois conjuntos tende-se a estabilizar próximo a época 10, o que caracteriza que não há ganhos significativos em se treinar o modelo por mais épocas.

B. Matriz de Confusão

As Tabelas III e IV apresentam as matrizes de confusão após o treinamento do modelo antes e depois da quantização, respectivamente.

Table III
MATRIZ DE CONFUSÃO DO MODELO NÃO QUANTIZADO.

	Manchadas	Ferrugem	Saudável
Manchadas	81,7%	15,0%	3,3%
Ferrugem	8,1%	90,5%	1,4%
Saudável	3,8%	1,9%	94,3%
F1 Score	0.84	0.89	0.94

Table IV
MATRIZ DE CONFUSÃO DO MODELO QUANTIZADO.

	Manchadas	Ferrugem	Saudável
Manchadas	85,0%	11,7%	3,3%
Ferrugem	13,5%	85,1%	1,4%
Saudável	5,7%	1,9%	92,5%
F1 Score	0.82	0.87	0.93

Um valor qualquer das Tabelas representa a porcentagem dos dados da i-ésima linha que foram classificados como dados da j-ésima coluna. Por exemplo, 15% dos dados da classe “Manchada” foram incorretamente classificados como “Ferrugem” no modelo não quantizado. A diagonal principal da tabela, destacadas em verde, representam os dados corretamente classificados.

O Gráfico da Figura 9 exibe o explorador de dados do modelo, complementando os dados da matriz de confusão da Tabela III do modelo não quantizado, observando-se, em vermelho, dados que foram erroneamente classificados entre as classes.

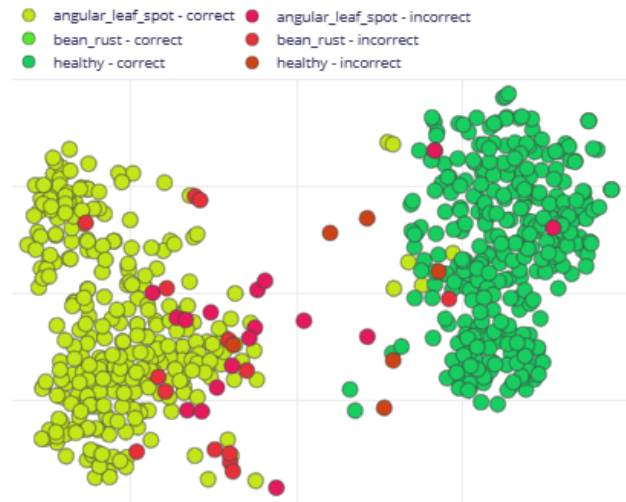


Figure 9. Data Explorer do modelo não quantizado.

É possível observar a presença da sobreposição das classes, e consequentemente, a ocorrência dos erros de classificação, com a dimensão da complexidade de definição de uma superfície de separação ideal para o problema enfrentado.

VI. APLICAÇÃO PRÁTICA

Esta seção detalha uma proposta de aplicação para o modelo, definida como “Monitoramento de Saúde de Culturas Agrícolas via Drones”, refletindo como o avanço tecnológico na área de drones e visão computacional oferece oportunidades significativas para a agricultura moderna.

A. Detalhamento da Aplicação

O sistema utiliza de câmeras em drones para filmar e avaliar as condições das folhas das plantas, identificando as doenças

mencionadas e auxiliando na tomada de ações preventivas. O drone sobrevoa a plantação, cobrindo toda a área cultivada, capturando imagens de forma sistemática.

B. Processamento de Imagens

As imagens capturadas são processadas pelo modelo treinado, que pode reconhecer as doenças “Ferrugem do Feijão” e “Mancha da Folha” nas folhas. A aplicação pode identificar áreas específicas onde as plantas mostram sinais de doenças ou outros problemas.

C. Benefícios

A detecção precoce de doenças nas plantas permite intervenções rápidas, reduzindo a propagação de doenças e minimizando perdas nas colheitas. Ao direcionar medidas corretivas apenas para áreas afetadas, os agricultores podem economizar recursos, como água e insumos agrícolas.

VII. CONCLUSÃO

A agricultura enfrenta desafios devido à doenças que afetam as culturas, impactando não apenas a produtividade, mas também a disponibilidade de alimentos para a população global. Com o aumento populacional e a crescente demanda por alimentos, torna-se obrigatório a exploração de tecnologias inovadoras, como a aprendizagem de máquina, para aumentar a eficiência na detecção e no controle de doenças nas plantações.

Nesse contexto, a obtenção de uma acurácia satisfatória pelo modelo de classificação proposto, assim como o baixo uso de recursos computacionais para a sua implementação, representa a definição de um potencial para o desenvolvimento de soluções que permitam a identificação precoce e precisa das doenças nas folhas do feijão. Ao reduzir as perdas nas colheitas, essa abordagem pode não apenas impulsionar a produtividade agrícola, mas também de contribuir significativamente para a segurança alimentar global, garantindo um suprimento mais estável e confiável de alimentos para a humanidade.

REFERENCES

- [1] Ministério da Agricultura e Pecuária, “Projeções do Agronegócio: Brasil - 2022/23 a 2032/33”, Governo Federal, 2022. Disponível em: <<https://www.embrapa.br/web/portal/soja/cultivos/soja1/dados-economicos>>. Acessado em: Outubro de 2023.
- [2] G. S. de C. Barros, L. R. A. Alves, A. L. R. Sanches, et al., “Efeito do não Tratamento de Pragas e Doenças Sobre Preços ao Consumidor de Produtos da Cadeia Produtiva de Soja”, pt. 3, Centro de Estudos Avançados em Economia Aplicada (CEPEA), Julho de 2019.
- [3] F. M. Lamas, “Artigo: A tecnologia na agricultura”, Embrapa, 2017. Disponível em: <<https://www.embrapa.br/busca-de-noticias/-/noticia/30015917/artigo-a-tecnologia-na-agricultura>>. Acessado em: Novembro de 2023.
- [4] R. L. A. Maranhão, J. E. R. V. Filho, “A Dinâmica do Crescimento das Exportações do Agronegócio Brasileiro.”, Ipea, 2016, p. 20.
- [5] Chen, J.; Liu, Q.; Gao, L. “Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model”. *Symmetry* 2019, 11, 343. Disponível em: <<https://doi.org/10.3390/sym11030343>>.
- [6] Liu, B.; Zhang, Y.; He, D.; Li, Y. “Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks”. *Symmetry* 2018, 10, 11. Disponível em: <<https://doi.org/10.3390/sym10010011>>.

- [7] Liang, Wj.; Zhang, H.; Zhang, Gf. et al. “Rice Blast Disease Recognition Using a Deep Convolutional Neural Network”. *Sci Rep* 9, 2869 (2019). Disponível em: <<https://doi.org/10.1038/s41598-019-38966-0>>.
- [8] M. Z. Siti, M. A. Zulkifley, M. M. Stofa, A. M. Kamari, and N. A. Mohamed, “Classification of tomato leaf diseases using MobileNetV2”, *Appl. Artif. Intell.*, vol. 9, no. 2, pp. 290–296, Jun. 2020. Disponível em: <<https://doi.org/10.11591/ijai.v9.i2.pp290-296>>.
- [9] M. S. Vinutha, R. Kharbanda, B. Rashmi, S. N. Rajani, and K. P. Rajesh, “CROP MONITORING: Using MobileNet models”, *Int. Res. J. England Tech.*, vol. 6, pp. 290–296, May 2019.
- [10] E. Elfatimi, R. Eryigit and L. Elfatimi, “Beans Leaf Diseases Classification Using MobileNet Models”, in *IEEE Access*, vol. 10, pp. 9471-9482, 2022, doi: 10.1109/ACCESS.2022.3142817.
- [11] TensorFlow Datasets Catalog, “Beans”. Disponível em: <<https://www.tensorflow.org/datasets/catalog/beans?hl=pt-br>>. Acessado em: Novembro de 2023.
- [12] Edge Impulse: Getting Started, “Edge Impulse Documentation”. Disponível em: <<https://docs.edgeimpulse.com/docs/>>. Acessado em: Novembro de 2023.
- [13] Edge Impulse: EON Tuner, “Edge Impulse Documentation”. Disponível em: <<https://docs.edgeimpulse.com/docs/edge-impulse-studio/eon-tuner>>. Acessado em: Novembro de 2023.
- [14] Keras: API reference - Keras Applications, “MobileNet, MobileNetV2 and MobileNetV3”. Disponível em: <<https://keras.io/api/applications/mobilenet/>>. Acessado em: Novembro de 2023.
- [15] Y. Du, C. L. Sershen, M. Wu, “Deep Learning Classification of Systemic Sclerosis Skin Using the MobileNetV2 Model”, in *IEEE Open Journal of Engineering*, 2021, p. 99.

ANEXOS

ANEXO A - Repositório no Github para o dataset e documentos referentes ao trabalho: <https://github.com/italotacca/proj_tp557>.