

Classificação de variedades de uma fruta: aplicação computacional em maçãs

Jonathan V. Kaizer^a, Rodrigo C. Thom de Souza^{a,c,d}, Linnyer B. R. Aylon^{a,b}, Yandre M. G. Costa^{a,b}

^aComputer Science Graduate Program (PCC), State University of Maringá (UEM), Maringá, PR, Brazil

^bDepartment of Informatics, State University of Maringá (UEM), Maringá, PR, Brazil

^cAdvanced Campus in Jandaia do Sul, Federal University of Parana (UFPR), Jandaia do Sul, PR, Brazil

^dProduction Engineering Graduate Program (PGP), State University of Maringá (UEM), Maringá, PR, Brazil

Resumo— Muitas vezes é difícil de fazer a distinção das variedades de uma fruta quando elas possuem características muito semelhantes. Esse é um problema encontrado nos caixas de supermercados, por exemplo, em que é preciso utilizar o reconhecimento visual para identificar variedades de maçãs, uma vez que cada variedade pode ter um preço diferente. Técnicas computacionais de classificação podem ser empregadas para resolver esse problema. Neste trabalho, foi utilizado Redes Neurais Convolucionais (CNN) com a arquitetura VGG-19 e técnica de *transfer-learning*, e uma abordagem com *K-Nearest Neighbors (K-NN)* utilizando $K=5$ para o número de vizinhos. A base escolhida continha 6 variedades de maçãs e foi adotado a estratégia de *undersampling* para balancear a base. Foi utilizado a estratégia de *cross-validation* para a validação dos resultados e a acurácia obtida pela CNN foi de $99,79 \pm 0,12\%$, e para o *K-NN* foi de $98,11 \pm 0,67\%$. Os resultados observados revelaram que o problema não tinha uma dificuldade tão elevada, o que trouxe altas acurácias.

Palavras-chave— CNN, K-NN, classificação, variedades, frutas

I. INTRODUÇÃO

Há uma crescente demanda na aplicação de reconhecimento e classificação de imagens. Esse processo consiste na extração de dados de uma imagem e comparação com um modelo computacional treinado [1]. A visão computacional tem sido muito utilizada nos processos de automação das indústrias e do varejo [12].

Uma das principais aplicações de classificações de imagens tem sido no reconhecimento de objetos, animais ou alimentos. Em produtos agrícolas, tem sido muito importante para vender produtos com preços adequados, de acordo com o nível de maturidade, tamanho ou tipo. Se fosse feito manualmente, esse processo seria inviável, custoso e cheio de erros. Portanto, é necessário recorrer a técnicas computacionais automatizadas [11].

Os estudos sobre classificação de frutas por imagem normalmente abordam a classificação de diferentes espécies, que possuem características bem distintas uma das outras [2]. No entanto, ela pode ser aplicada também para classificar variações de uma mesma espécie de fruta, como a maçã gala e a maçã argentina. As principais características utilizadas nestes modelos tem sido as cores em *Red-Green-Blue (RGB)*, texturas e a forma do objeto [11] [13].

Muitas vezes é difícil de fazer a distinção das variedades de uma fruta quando elas possuem características muito semelhantes, como forma, cor e tamanho. Esse é um problema encontrado nos caixas de supermercados, por

exemplo, em que o operador de caixa tem de utilizar o reconhecimento visual para identificar a fruta e a sua variedade, uma vez que cada variedade pode ter um preço diferente [4], o que torna a operação passível de erros. A classificação pode ser utilizada também para substituir a necessidade de código de barras nos produtos [12].

Logo, a motivação deste estudo é auxiliar as pessoas na correta identificação de variedades de determinada fruta de forma rápida e eficiente, contribuindo com a cadeia de suprimento de alimentos frescos ao reduzir custos e reduzir a carga de trabalho [13].

Este estudo tem como objetivo avaliar a aplicação de métodos computacionais na classificação de variedades da maçã, a partir de uma base de imagens obtida no *Kaggle*. Para isso, foi utilizado Redes Neurais Convolucionais (CNN) e uma abordagem com *K-Nearest Neighbors (K-NN)*. Desta forma, é possível comparar e utilizar essas abordagens e implementar nas operações diárias da indústria e comércio de alimentos frescos.

O restante deste artigo está organizado da seguinte forma: na Seção II é apresentado os trabalhos relacionados, na Seção III é apresentado a metodologia empregada e os modelos construídos, na Seção IV são apresentados os resultados da execução computacional dos modelos apresentados, e por fim, na Seção V são apresentadas as conclusões do estudo.

II. TRABALHOS RELACIONADOS

Existem vários métodos utilizados na literatura para aplicações de visão computacional e classificação.

Malinao e Hernandez [3] utilizam CNN para classificar árvores da espécie *Artocarpus*. Os dados consistiam em 2700 imagens igualmente distribuídas em 3 classes, sendo que cerca de 10% dessas imagens foram utilizadas para validação dos resultados e outros 20% foram utilizados para testes. Os testes demonstraram cerca de 82 a 94% de acurácia. O estudo destaca que muitos estudos utilizam CNN para classificar plantas, árvores e plantações, uma vez que possuem acurácia melhor do que as redes neurais artificiais (ANN).

Cizotto *et al.* [16] avaliaram protótipos de páginas Web usando CNN e *Class Activation Mapping (CAM)* com validação cruzada com 10 *folds*, obtendo acurácia de 95.71% para o melhor modelo.

Hossain *et al.* [4] abordam a classificação de frutas diferentes. A classificação precisa de frutas é um tópico importante e relevante para aplicações industriais. O principal

problema descrito é que normalmente os produtos são identificados por códigos de barras nos supermercados, mas isso não ocorre com as frutas vendidas por eles. O trabalho contribui com a correta identificação e classificação da fruta, sendo um auxílio para a correta detecção e faturamento da venda. Utilizaram duas bases de dados: a primeira com 15 classes contendo cerca de 2633 imagens e a segunda base com 10 classes contendo 5946 imagens, e cada classe representava um tipo de fruta, como laranja e maçã. O destaque da metodologia empregada fica pela utilização de uma *CNN* com o modelo pré-treinado *VGG-16*. Os resultados mostraram uma acurácia acima de 99% para a primeira base e de cerca de 96% para a segunda.

Muladi *et al.* [11] aplicaram uma rede neural para classificar a qualidade de maçãs, com a finalidade de detectar pela cor se a fruta estava adequada ou não para consumo. Para isso, a *feature* extraída e utilizada foi a cor *RGB* obtida em imagens capturadas da fruta. A cor é um bom indicador de tipo, frescor e qualidade. Para tanto, utilizaram cerca de 20 imagens que tiveram que ser segmentadas para remover o plano de fundo e aumentar a acurácia do modelo. Mesmo com a pequena quantidade de dados, os resultados mostraram se altos e por volta de 90%.

Kumar *et al.* [5] abordam em seu trabalho a classificação de patógenos em folhas da árvore da fruta manga, utilizando uma *CNN* inspirada na arquitetura do modelo pré-treinado *VGG-16*. A abordagem é realizada utilizando 80% das 3500 imagens para treinamento e o restante para validação. Embora tenham usado 100 épocas, analisando os gráficos de acurácia e perda é possível ver que eles se estabilizam por volta da 5 a 10 época.

Satake *et al.* [6] abordam o uso de *CNN* para reconhecimento de 9 classes de plantas tóxicas. Realizaram um pré-processamento onde removeram imagens muito similares, depois realizaram um *data augmentation*. A utilização dos modelos pré-treinados ocorreu usando o banco de dados da *ImageNet*, que consiste em milhões de imagens separadas em 22 mil categorias, o que proporciona uma melhor generalização para o modelo a ser utilizado e o treinamento em poucas épocas. O *VGGNet*, que envolve o *VGG-16* e *VGG-19*, é uma arquitetura para redes profundas amplamente utilizado para classificação de imagens. Outras arquiteturas usadas foram a *DenseNet*, *Resnet* e *Xception*. Além disso, foi utilizado validação cruzada com 10 *folds* e os modelos foram treinados com 10 épocas.

Nakahata *et al.* [7] classificam tipos de Bonsai com uma base de 700 imagens e 7 classes. Para isso, usaram *CNN* comparando arquiteturas como a *VGG*, *Xception*, *DenseNet* e *Inception*. Usou também *transfer learning* com os pesos pré-treinados da base de dados *ImageNet*, pois essa técnica ajuda a melhorar o desempenho quando a base de dados é pequena. Recentemente muitos estudos têm utilizado *CNN* combinados com *transfer learning*. As imagens julgadas muito similares foram removidas e houve a aplicação de *data augmentation* para evitar a ocorrência de *overfitting*. Também usou validação cruzada, que consiste em cada estágio utilizar uma parte diferente dos dados para treinamento e validação.

CNN é uma técnica de *Deep Learning* considerada uma das melhores formas de reconhecimento de imagens [5], podendo ser supervisionado ou não supervisionado. Ela consiste em um modelo complexo com várias camadas de processamento, sendo a principal vantagem a não necessidade

de extrair atributos manualmente. As *CNN* são modelos muito mais rígidas e é inspirado na estrutura córtex visual do cérebro, tendo provado a sua eficácia ao classificar em grande escala as imagens da *ImageNet* e a capacidade de reaproveitar modelos pré-treinados para outros problemas [3].

Nyon *et al.* [1] abordam o uso do *K-NN* para classificar as variações da fruta Durian (uma fruta cultivada na Malásia). O *K-NN* calcula as distâncias euclidianas entre as imagens do treinamento e do teste. O número de vizinhos considerados em cada um desses cálculos deve ser especificado manualmente através de testes empíricos. Foram utilizados para treinamento e validação cerca de 40 imagens, 10 imagens para cada uma das 4 espécies. Utilizando-se de *cross-validation*, os autores dividiram a validação em 10 partes e conseguiram obter uma acurácia de 100%.

Yadav *et al.* aplicam o *K-NN* na classificação de corpos celestes, considerado amplamente utilizado para dados esparsamente densos. Embora não tenha relação com produtos alimentícios, a metodologia pode ser utilizada como base em trabalhos envolvendo classificação. No estudo, o histograma de cores foi utilizado como *feature* para ser aplicado à classificação, transformando as matrizes de cores *RGB* em um vetor com apenas uma dimensão. Foi utilizado também a estratégia de *cross-validation* que consiste em dividir os dados em *k-folds*, tomando uma parte dessa para validação e as demais para treinamento do modelo. O *K-NN* utilizando *cross-validation* trouxe resultados consistentes e com boa acurácia para o problema proposto.

Zawbaa *et al.* [12] classificam três frutas diferentes com cores muito semelhantes, utilizando o *K-NN*, *Random Forest (RF)* e *Support Vector Machine (SVM)*. As amostras consistiam em 178 imagens que foram reduzidas para o tamanho 90x90.

Vaishnavel *et al.* [8] aplicam o *K-NN* para detecção de doenças nas folhas da planta do amendoim, uma das principais culturas cultivadas e importantes para a Índia. Há 4 classes de doenças que foram treinadas e detectadas com um conjunto de 250 imagens. No reconhecimento de padrões, o *K-NN* é um método não paramétrico em que é realizado uma análise e classificação para os *K* vizinhos mais próximos de cada imagem. O *K-NN* é considerado um dos mais simples modelos de aprendizagem de máquina.

III. METODOLOGIA

Para o desenvolvimento deste estudo, foram utilizados uma base de dados do *Kaggle*¹ com variações de espécie de maçãs. A base escolhida continha 6 variedades separadas. Cada variedade foi considerada como uma classe, uma vez que o objetivo é classificar as variedades de maçã. A quantidade de imagens para cada classe pode ser visualizada na Tabela 1.

Tabela 1

IMAGENS POR CLASSE DA BASE ORIGINAL

Classe	Quantidade
Apple A	692
Apple B	740
Apple C	1002
Apple D	1033
Apple E	664
Apple F	2030

¹ <https://www.kaggle.com/chrisfilo/fruit-recognition>

As imagens obtidas mostram uma variedade por imagem, sendo que há uma ou mais maçãs da mesma variedade sob uma bandeja prateada, conforme Figura 1. As maçãs D e F possuem uma maior similaridade entre elas, assim como a A e C. A maçã do tipo B possui um formato alongado característico que diferencia ela das demais.

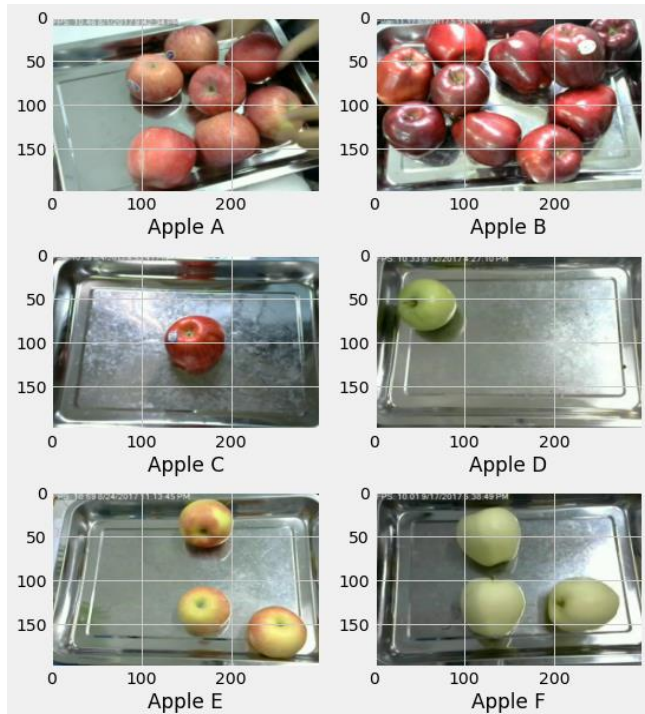


Figura 1. Exemplos de variedades de maçã separadas por classes de A à F. As imagens não foram segmentadas e é possível notar um certo padrão no plano de fundo caracterizado por uma bandeja prateada. O número de maçãs também varia para cada imagem, mas cada imagem contém apenas uma variedade.

Para o desenvolvimento do modelo, foi utilizada a linguagem de programação *Python* e as bibliotecas *OpenCV*, *Keras*, *TensorFlow* e *Scikit-learn*. O *Python* associado com estas bibliotecas tem sido muito utilizada para resolver problemas de *Machine Learning*. A codificação e execução ocorreram na plataforma web do *Kaggle*, utilizando os recursos de *Central Processing Unit (CPU)* e *Graphics Processing Unit (GPU)* disponibilizadas pela empresa.

Como a base estava desbalanceada, foi adotado a estratégia de *undersampling* onde foram selecionadas 664 imagens de cada classe de forma aleatória. O *undersampling* consiste em pegar amostras de forma aleatória, limitada a quantidade da classe com menor quantidade [14]. A nova quantidade para cada classe pode ser vista na Tabela 2.

Tabela 2
IMAGENS POR CLASSE DA BASE BALANCEADA

Classe	Quantidade
Apple A	664
Apple B	664
Apple C	664
Apple D	664
Apple E	664
Apple F	664

Após obter a base, as imagens foram carregadas e alocadas na memória por meio da biblioteca *OpenCV*. Constatou-se uma variação no tamanho das imagens da base e por isso, todas foram redimensionadas. O tamanho das imagens armazenadas na memória ficou em $150 \times 150 \times 3$. Não foram aplicadas outras técnicas de pré-processamento nas imagens, mas é possível remover o *background*, ajustar a iluminação ou isolar apenas uma maçã por amostra.

Para o *CNN*, as imagens foram normalizadas e os valores de *RGB* foram convertidos para valores entre 0 e 1. Já para o *K-NN*, a matriz de cada imagem foi convertida para um vetor com apenas uma dimensão, através da função “*Flatten*”, ou seja, cada imagem era representada por um vetor de única dimensão contendo todos os valores de *RGB*. Em seguida, o vetor foi normalizado através do componente *StandardScaler* do *scikit-learn*. O *StandardScaler* normaliza os dados ao escalonar de forma que a média seja zero e com um desvio padrão de 1 [9].

A estratégia de validação dos resultados escolhida foi a de *cross-validation*. A execução do modelo foi dividida em 5 *k-folds*, o que consistia em executar o modelo 5 vezes pegando 1/5 da base para validação (20%), sendo que a cada execução esse pedaço era diferente. Um dos cuidados foi em construir *fold* que continham a mesma quantidade de amostras por classe. Foi utilizado a estratégia de *K-fold* estratificado, que tem como objetivo garantir que o *fold* de teste tenha uma quantidade igual de imagens para cada classe [15]. A Tabela 3 exemplifica esse processo. Além disso, antes de realizar a divisão dos dados, foi realizado uma mistura aleatória das imagens, com o objetivo de distribuir melhor e evitar que os *folds* contivessem um conjunto tendencioso, ou seja, garantiu-se que cada *fold* contivesse uma amostra aleatória das imagens de cada classe e que o conjunto de validação contivesse amostras que representariam todas as demais da base de dados.

Tabela 3
EXEMPLO DA DIVISÃO DOS DADOS EM *K-FOLDS*

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Iteração 1	Teste	Treino	Treino	Treino	Treino
Iteração 2	Treino	Teste	Treino	Treino	Treino
Iteração 3	Treino	Treino	Teste	Treino	Treino
Iteração 4	Treino	Treino	Treino	Teste	Treino
Iteração 5	Treino	Treino	Treino	Treino	Teste

Na Figura 2, é possível ver os passos adotados no desenvolvimento dos classificadores. Tendo já abordado a aquisição dos dados, o pré-processamento e o *cross-validation*, será abordado em seguida a metodologia empregada na construção do modelo *CNN* e *K-NN*, ambos utilizados neste trabalho para fins de comparação. Vale ressaltar que os modelos foram executados separadamente e, por isso, os dados separados nos *folds* pelo *cross-validation* não eram idênticos para os dois modelos.

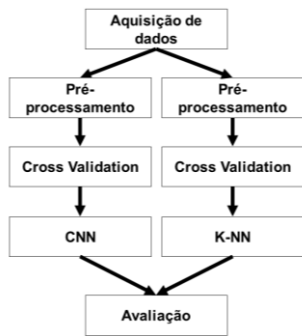


Figura 2. Representação dos passos seguidos para a construção dos classificadores.

A. CNN

A técnica de *transfer-learning* foi utilizada com a arquitetura *VGG-19*, por simplificar a aprendizagem e classificação. O *VGG-19* foi pré-treinado utilizando o banco de imagens da *ImageNet*.

No modelo foi adicionado a camada pré-treinada e em seguida foi adicionado a camada “*Flatten*”, que tem como função transformar os dados para um vetor de uma dimensão. Depois, uma camada densa com 128 neurônios foi utilizada com o ativador *Rectified Linear Unit (ReLU)*, assim como um *dropout* de 0,3 para ajudar a evitar super adaptação [4]. Em seguida, uma camada densa com 6 neurônios – que é o número de saídas – foi utilizada em conjunto com o ativador *Softmax*. Essa função de ativação classifica as saídas em valores entre 0 e 1, representando a probabilidade para cada classe. A arquitetura final está representada na Figura 3.

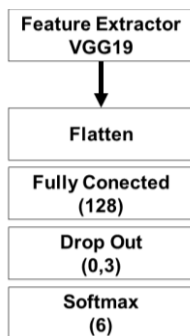


Figura 3. Arquitetura final da *CNN*. Os parâmetros estão apresentados em parênteses. Inspirado em [7].

O modelo foi compilado usando o otimizador *Adam*, monitorando a acurácia. Por fim, para treinamento do modelo, 10 épocas foram suficientes para cada uma das 5 iterações do *k-fold*.

B. K-NN

O *K-NN* pode ser implementado facilmente ao invocar a função “*KNeighborsClassifier*” do *scikit-learn*. Com os dados de entrada normalizados e as identificações de cada imagem definidas, o modelo foi compilado e treinado para cada uma das 5 iterações do *k-fold*, como forma de mensurar melhor a acurácia. Além disso, o número de vizinhos utilizado no modelo foi de 5, valor normalmente utilizado na literatura [1] e valor padrão sugerido pela biblioteca utilizada. Os outros parâmetros foram empregados com valores padrões, tais como o uso de pesos uniformes, a escolha automática do algoritmo (*ball_tree*, *kd_tree* ou *brute*) e a utilização da métrica *minkowski* para calcular as distâncias.

IV. RESULTADOS E DISCUSSÕES

Um exemplo dos resultados da validação da classificação pode ser observado na Figura 4, onde é apresentada a imagem, a classe representada e a classe que foi predita.

A. Resultados da CNN

A acurácia obtida foi de 99,79% ao fazer uma média das acurácias de cada iteração do *k-fold*, mais um desvio-padrão de 0,12%, conforme Tabela 4. O tempo de execução foi de cerca de 14 minutos, tendo sido utilizado a *GPU* do *Kaggle*.

Tabela 4

ACURÁCIA E PERDA EM CADA ITERAÇÃO DE TREINAMENTO

<i>Fold</i>	<i>Loss</i>	<i>Accuracy</i>
1	0.02090	99,12%
2	0.01102	99,62%
3	0.00920	99,62%
4	0.01581	99,25%
5	0.01083	99,62%



Figura 4. Exemplo dos resultados das predições. A análise visual da predição das imagens foi realizada para se certificar que a imagem, a categoria original e a categoria prevista estavam condizentes e que não havia nenhum erro de codificação do algoritmo.

A Figura 5 representa a acurácia encontrada para o treino e para a validação da iteração que obteve o melhor resultado. É possível notar que a precisão do treinamento e da validação sobem logo nas primeiras épocas. De forma análoga, o erro do treinamento e da validação caem ao longo das primeiras iterações. A acurácia da validação se mostrou maior que o do treinamento devido ao uso de um *dropout* mais elevado, que faz com que alguns neurônios não sejam ativados durante o treinamento.

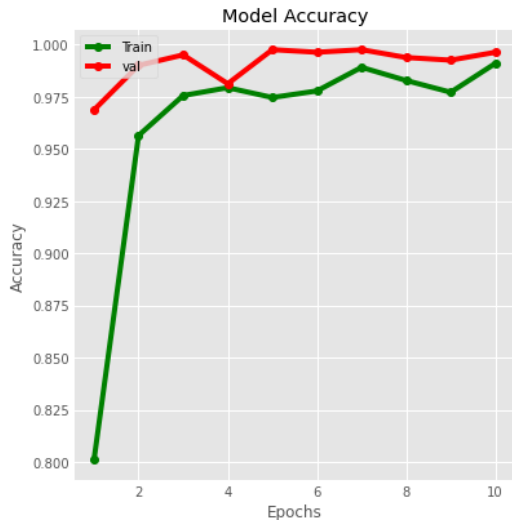


Figura 5. Gráfico da evolução da acurácia conforme as épocas de treinamento.

Da mesma forma, o *loss* registrado na validação foi menor do que no treinamento devido ao uso do *dropout*, conforme pode ser visto na Figura 6. Se interpretarmos o gráfico, o modelo treinado consegue generalizar muito bem a classificação dos dados de validação.



Figura 6. Gráfico da evolução da perda conforme as épocas de treinamento.

Para avaliar os resultados do modelo, é comum encontrar na literatura o uso da matriz de confusão. A matriz de confusão relaciona a base de teste com a base de validação ao comparar o que foi predito e o que deveria ser. Neste caso, foi construído uma matriz de confusão com a soma dos resultados das validações de cada iteração do *k-fold*, conforme Figura 7.

B. Resultados do K-NN

O modelo *K-NN* foi executado com um número de vizinhos $K=5$ e o tempo de execução foi de 64 minutos, tendo sido utilizado apenas a *CPU* para o treinamento. A acurácia obtida foi de 98,11% ao fazer uma média das acurácias de cada iteração do *k-fold*, mais um desvio-padrão de 0,67%, conforme Tabela 5.

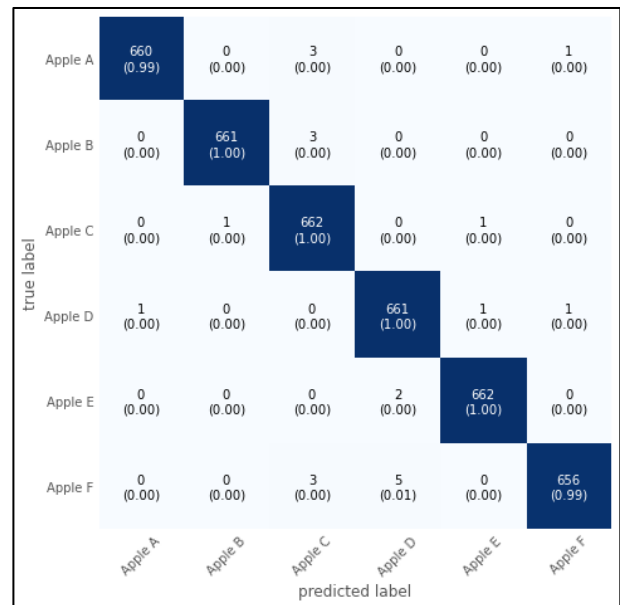


Figura 7. Matriz de confusão para o modelo CNN

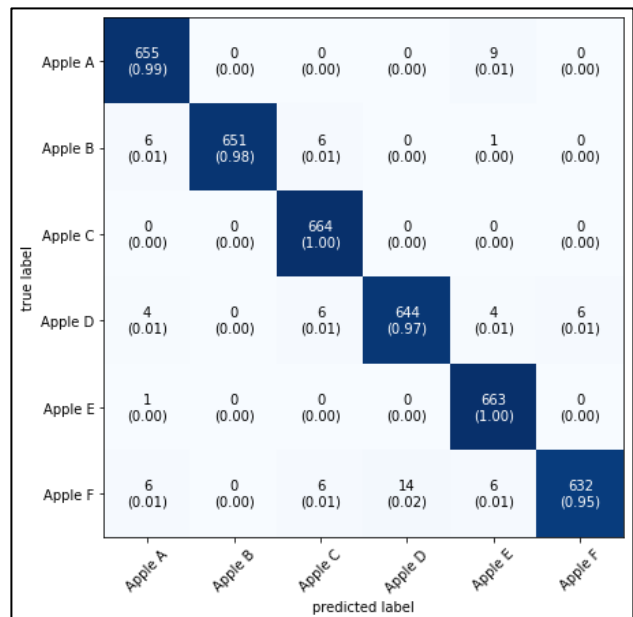


Figura 8. Matriz de confusão para o modelo K-NN

Tabela 5

ACURÁCIA EM CADA ITERAÇÃO DE TREINAMENTO

Fold	Accuracy
1	97,99 %
2	98,49%
3	96,86%
4	98,74%
5	98,49%

A matriz de confusão relaciona a base de teste com a base de validação ao comparar o que foi predito e o que deveria ser. Assim como no modelo anterior, foi construído uma matriz de confusão com a soma dos resultados das validações de cada iteração do *k-fold*, conforme Figura 8.

Analisando os resultados, o modelo de *CNN* se mostrou com maior acurácia e gastou consideravelmente menos tempo que o *K-NN*, principalmente porque a *CNN* utiliza da *GPU* enquanto o *K-NN* apenas a *CPU*. Seria mais justo comparar o tempo treinando ambos os modelos utilizando a *GPU*, mas neste estudo o foco foi analisar a acurácia.

A abordagem de *cross-validation* se mostrou importante por revelar que há variações na acurácia de acordo com a porção de dados escolhida para validação, mesmo que nos resultados encontrados não se mostrou tão significativo.

Uma inspeção visual dos resultados preditos foi feita comparando com a imagem e a categoria inicial para se certificar que as imagens estavam de acordo com as categorias de treinamento e que a categoria predita estava condizente com a categoria real na maioria dos resultados, conforme abordado na Figura 3.

V. CONCLUSÕES

Neste trabalho foi avaliado a classificação de variedades de maçãs utilizando uma arquitetura de *CNN* com *transfer learning* e uma abordagem com *K-NN*. Os experimentos foram realizados utilizando-se de uma base de imagens disponível na internet e acessível por qualquer um.

Os experimentos demonstraram que a *CNN* com a arquitetura *VGG-19* possui a melhor acurácia na classificação de imagens de variedades de maçãs. Poderia ter sido aplicada também a estratégia de *data augmentation* para aumentar o número de amostras, pois modelos *CNN* necessitam de grandes quantidades de imagens para evitar o efeito da super aprendizagem (*overfitting*). No entanto, esse problema foi contornado graças ao alto *dropout* utilizado para evitar a super adaptação dos treinamentos, o que trouxe resultados adequados e um modelo generalizável. Outro motivo de não ter utilizado *data augmentation* foi para fazer uma comparação equivalente entre os modelos *CNN* e *K-NN*, utilizando-se da mesma quantidade de dados.

Em contrapartida, o modelo utilizando *K-NN* obteve também uma boa acurácia, mas inferior ao do *CNN*. Outro ponto de importante de destaque é o tempo computacional gasto que ficou quase 5x superior ao do *CNN*, mas isso se deve ao fato de que este modelo utilizou apenas a *CPU* para o treinamento e essa comparação de tempo pode não ser tão justa assim. O modelo utilizou-se de 5 vizinhos para o treinamento, mas o teste com outras quantidades seria recomendado para avaliar os resultados.

Embora cada imagem tivesse um número aleatório de unidades de maçã, a contagem dessa quantidade não foi abordada e os modelo empregados conseguiram se adaptar ao ponto de a quantidade não importar. De forma geral, os resultados foram bons e ambos os modelos se mostraram de boa acurácia, com destaque para o *CNN*, mas com eficiência e tempo computacional significativamente diferentes entre os modelos.

Os modelos analisados se mostraram adequados, mas os resultados observados pelo gráfico de acurácia e épocas do modelo *CNN* revelaram que o problema não tinha uma

dificuldade tão elevada como era esperado ao observar as imagens originais, o que trouxe acurácias muito altas, até mesmo para o modelo *K-NN* que geralmente tem resultados significativamente inferiores. Diante desse fato, o algoritmo e a divisão dos *folds* foram revisados na esperança de encontrar vazamentos ou problemas que justificassem os resultados, mas nada foi encontrado. Logo, a conclusão que se chegou era de que o problema não tinha um desafio tão significativo e que é preciso uma análise mais detalhada da base de imagens para detectar alguma anormalidade, como a presença de imagens muito semelhantes. Visando tornar o problema mais desafiador, como trabalhos futuros pretende-se ainda utilizar um conjunto de imagens maior para classificação das variedades de maçãs, se possível com diversas variedades de maçãs em uma mesma imagem, bem como utilizar outras técnicas e métricas de avaliação para o problema.

AGRADECIMENTOS

Agradecemos o Departamento de Informática (DIN) da Universidade Estadual de Maringá (UEM), o CNPq (311685/2017-0 e 163961/2021-2) e Fundação Araucária (17.633.124-0) pelo suporte financeiro. O presente trabalho também foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

- [1] X. Y. Nyon, M. R. Mustaffa, L. N. Abdullah and N. A. Nasharuddin, "Durian Species Recognition System Based on Global Shape Representations and K-Nearest Neighbors," 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), 2018, pp. 1-6, doi: 10.1109/INFRKM.2018.8464795.
- [2] M. Senthilarasi, S. M. M. Roomi and M. R. H. Prasanna, "Shape based approach for detecting Musa Species in fruit industry," 2014 Sixth International Conference on Advanced Computing (ICoAC), 2014, pp. 157-160, doi: 10.1109/ICoAC.2014.7229765.
- [3] R. M. L. Malinao and A. A. Hernandez, "Artocarpus Trees Classification using Convolutional Neural Network," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2018, pp. 1-6, doi: 10.1109/HNICEM.2018.8666322.
- [4] M. S. Hossain, M. Al-Hammadi and G. Muhammad, "Automatic Fruit Classification Using Deep Learning for Industrial Applications," in IEEE Transactions on Industrial Informatics, vol. 15, no. 2, pp. 1027-1034, Feb. 2019, doi: 10.1109/TII.2018.2875149.
- [5] P. Kumar, S. Ashtekar, S. S. Jayakrishna, K. P. Bharath, P. T. Vanathi and M. Rajesh Kumar, "Classification of Mango Leaves Infected by Fungal Disease Anthracnose Using Deep Learning," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1723-1729, doi: 10.1109/ICCMC51019.2021.9418383.
- [6] S. S. Satake, R. Calvo, A. S. Brito Jr., Y. M. G. Costa, "Classification of Toxic Ornamental Plants for Domestic Animals using *CNN*", 2021, unpublished.
- [7] G. H. S. Nakahata, A. A. Constantino and Y. M. G. Costa, "Bonsai Style Classification: a new database and baseline results," 2020 IEEE International Symposium on Multimedia (ISM), 2020, pp. 104-110, doi: 10.1109/ISM.2020.00025.
- [8] M. P. Vaishnave, K. S. Devi, P. Srinivasan and G. A. P. Jothi, "Detection and Classification of Groundnut Leaf Diseases using KNN classifier," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 2019, pp. 1-5, doi: 10.1109/ICSCAN.2019.8878733.

- [9] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi and V. Padma, "Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 729-735, doi: 10.1109/ICSSIT48917.2020.9214160.
- [10] S. Yadav, A. Kaur and N. S. Bhauryal, "Resolving the celestial classification using fine K - NN classifier," 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2016, pp. 714-719, doi: 10.1109/PDGC.2016.7913215.
- [11] M. Muladi, D. Lestari and D. T. Prasetyo, "Classification Of Eligibility Consumption Of Manalagi Apple Fruit Varieties Using Backpropagation," 2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA), 2019, pp. 75-79, doi: 10.1109/ICAMIMIA47173.2019.9223398.
- [12] H. M. Zawbaa, M. Hazman, M. Abbass and A. E. Hassanien, "Automatic fruit classification using random forest algorithm," 2014 14th International Conference on Hybrid Intelligent Systems, 2014, pp. 164-168, doi: 10.1109/HIS.2014.7086191.
- [13] A. Kanade and A. Shaligram, "Development of machine vision based system for classification of Guava fruits on the basis of CIE1931 chromaticity coordinates," 2015 2nd International Symposium on Physics and Technology of Sensors (ISPTS), 2015, pp. 177-180, doi: 10.1109/ISPTS.2015.7220107.
- [14] J. Xue and P. Hall, "Why Does Rebalancing Class-Unbalanced Data Improve AUC for Linear Discriminant Analysis?," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 5, pp. 1109-1112, 1 May 2015, doi: 10.1109/TPAMI.2014.2359660.
- [15] M. A. Alim, S. Habib, Y. Farooq and A. Rafay, "Robust Heart Disease Prediction: A Novel Approach based on Significant Feature and Ensemble learning Model," 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2020, pp. 1-5, doi: 10.1109/iCoMET48670.2020.9074135.
- [16] Cizotto, A.A.J., de Souza, R.C.T., Mariani, V.C. et al. Web pages from mockup design based on convolutional neural network and class activation mapping. *Multimed Tools Appl* (2023). <https://doi.org/10.1007/s11042-023-15108-3>