

# Mineração de Imagens - Identificação de doenças em culturas agrícolas

Henrique Gomes Zanin - 10441321

<sup>1</sup>Relatório desenvolvido para a  
Disciplina de Mineração de Dados não Estruturados - ICMC-USP

**Resumo.** *O trabalho explora a aplicação de técnicas de redes neurais convolucionais (CNN) e ResNet para classificação de doenças culturas agrícolas, abordando um problema relevante para o setor agrícola brasileiro. Utilizando um conjunto de imagens rotuladas, são testadas diferentes abordagens de pré-processamento e estratégias de balanceamento para otimizar a acurácia dos modelos. O modelo CNN com Early Stopping no conjunto desbalanceado se destaca, mostrando melhor desempenho. A técnica de KNN aplicada nas embeddings geradas pelos modelos também apresentou bons resultados quando comparada com a classificação dos modelos. Os resultados mostram que, embora os métodos utilizados não ultrapassem um F1-Score ponderado de 0,64, a utilização de técnicas adicionais de segmentação e eliminação de ruído poderia potencialmente melhorar a eficácia dos classificadores.*

## 1. Identificação do Problema

O agronegócio no Brasil respondeu em média por 23% do PIB<sup>1</sup> entre os anos de 1995 e 2021. Diante da elevada importância do setor na economia brasileira, a manutenção da cadeia produtiva mostra-se relevante para a garantia de crescimento futuro dos agregados macroeconômicos.

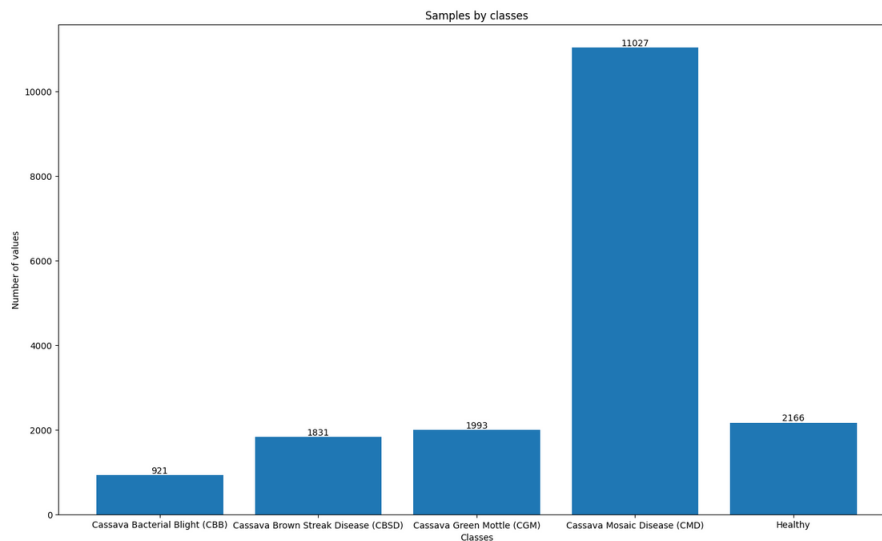
Um dos maiores desafios inerentes à perda de produtividade das lavouras são as doenças associadas às espécies agrícolas. A identificação precoce de patógenos nas plantas colabora para a aplicação de medidas de contenção na propagação da doença para os exemplares saudáveis.

Como forma de colaborar para a identificação antecipada de doenças em lavouras, este trabalho propõe um classificador de imagens para avaliar a saúde das espécies. Utilizando um conjunto de imagens rotuladas previamente extraído da página <https://www.kaggle.com/datasets/mexwell/crop-diseases-classification/data> avaliamos a acurácia de rede neurais convolucionais e de técnicas de classificação associadas às embeddings extraídas do modelo para detecção de doenças.

O conjunto de dados apresenta 5 categorias para classificação: "Cassava Bacterial Blight (CBB)", "Cassava Brown Streak Disease (CBSD)", "Cassava Green Mottle (CGM)", "Cassava Mosaic Disease (CMD)", e "Healthy". As amostras de acordo com cada classe estão distribuídas de acordo com o gráfico exibido na Figura 1

---

<sup>1</sup><https://cepea.esalq.usp.br/br/opiniao-cepea/afinal-quanto-o-agronegocio-representa-no-pib-brasileiro.aspx>

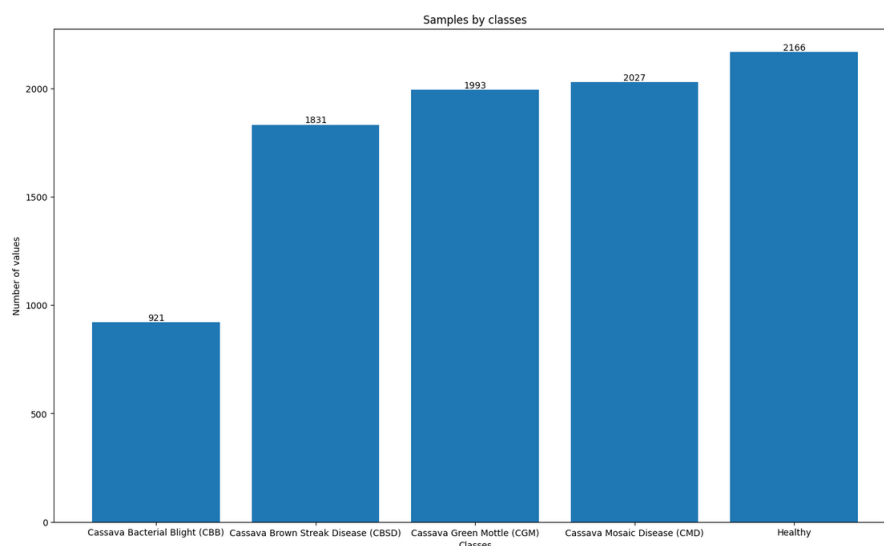


**Figura 1. Conjunto de dados**

## 2. Pré-processamento

O conjunto de dados original contém um arquivo csv que indica a classe de cada imagem e as imagens a serem processadas. No arquivo csv consta uma quantidade de imagens superior às existentes, dessa forma realizou-se um filtro no csv para remover os dados inválidos.

Para a correta avaliação dos modelos foi a realização de um undersampling manual na classe Cassava Mosaic Disease (CMD) para que contenha um número de amostras próximo às demais, resultando na distribuição exibida na Figura 2:



**Figura 2. Conjunto de dados com undersampling**

Foram aplicadas duas transformações no conjunto de dados. A primeira foi o redimensionamento das imagens de 600 X 800 para 128 X 128 para a CNN e 224 X 224 para a ResNet18. O modelo ResNet18 exige o redimensionamento para 224 X 224 por possuir

como conjunto de treinamento original imagens nesse formato. A segunda transformação consiste na aplicação de uma normalização nos testes envolvendo a CNN com Early Stopping (ES) e a ResNet18.

### 3. Extração de Padrões

Para a classificação das imagens foram utilizadas 3 abordagens como descritas a seguir:

- **CNN sem Early Stopping:** Treinamento da CNN e aplicação das embeddings no KNN
- **CNN com Early Stopping:** Treinamento da CNN com um early stopping com o parâmetro "patience" definido como 10 (finaliza o treinamento caso o loss no conjunto de validação não melhore em até 10 épocas). Após o treinamento utilizou-se as embeddings para aplicação do KNN
- **ResNet18 com Early Stopping:** Treinamento da última camada da ResNet pré-treinada e aplicação do KNN nas embeddings da ResNet. Utilizou o mesmo parâmetro de early stopping da CNN.

Cada uma das abordagens foram executadas no conjunto balanceado e desbalanceado para testar a acurácia da classificação. Contando com a execução do KNN nas embeddings geradas pelos modelos, totaliza-se 12 testes de classificação (2 conjuntos de dados \* (embeddings com knn + classificação com modelo) \* 3 modelos). Todos os modelos foram comparados com os respectivos classificadores Dummy.

Os modelos com a aplicação de early stopping foram salvos com o objetivo de recuperar o que obteve a melhor desempenho. Quando a execução finaliza-se devido ao critério de parada o melhor modelo é salvo no diretório em que o notebook encontra-se.

Os conjuntos de treino, teste e validação (quando há early stopping) foram separados da seguinte forma:

- **Sem Early Stopping:**
  - Treino: 80%
  - Teste: 20%
- **Com Early Stopping:**
  - Treino: 56%
  - Teste: 20%
  - Validação: 24%

O conjunto de validação foi definido com sendo 30% do conjunto de treino. Assim, como o conjunto de treino sem validação continha 80% das amostras, temos  $0.8 \times 0.3$ , resultando no valor de 0.24 do conjunto total.

Para classificação utilizou-se dois métodos. O primeiro consiste aplicação do modelo treinado nas imagens do conjunto de teste, originando tensores para cada imagem. Os tensores resultantes da saída do modelo são entradas para a função *max* do pacote pytorch que determina a probabilidade de pertencimento de uma imagem para as classes. O comportamento da função *max* é similar a da função *argmax*, com exceção ao fato da *max* ser paralelizável em núcleos CUDA

O segundo método utiliza o vetor de embeddings, construído a partir do modelo treinado, para a classificação por meio do KNN. Os vetores de embeddings gerados para o

KNN são também utilizados para o classificador Dummy. Como estratégia de otimização do KNN há uma função que seleciona o melhor valor de k para 50 execuções do algoritmo

2

#### 4. Pós-processamento

Os resultados obtidos através dos testes demonstraram que o balanceamento não surtiu efeitos positivos na acurácia dos modelos. Em todos os testes o F1-Score poderado foi inferior no conjunto de dados balanceado quando comparados com o conjunto balanceado. De fato, a estratégia de balanceamento manual que resultou em um undersampling de uma classe não foi eficaz. A Tabela 1 sumariza dos resultados dos testes.

**Tabela 1. Resultados compilados dos métodos de aprendizado supervisionado**

Métodos	Parâmetros	Epocas	F1-Score/Dummy
CNN sem balanceamento	lr = 0.001	30	0.58 / 0.42
<b>KNN - Embeddings CNN sem balanceamento</b>	<b>k = 42</b>	-	<b>0.62 / 0.42</b>
CNN com balanceamento	lr = 0.001	30	0.37 / 0.21
KNN - Embeddings CNN com balanceamento	k = 15	-	0.38 / 0.21
<b>CNN com ES sem balanceamento</b>	<b>lr = 0.001</b>	<b>13</b>	<b>0.64 / 0.20</b>
KNN - Embeddings CNN com ES sem balanceamento	k = 17	-	0.50 / 0.20
CNN com ES com balanceamento	lr = 0.001	13	0.39 / 0.20
KNN - Embeddings CNN com ES com balanceamento	k = 49	-	0.41 / 0.20
ResNet sem balanceamento	lr = 0.001	20	0.11 / 0.22
KNN - ResNet sem balanceamento	k = 48	-	0.54 / 0.22
ResNet com balanceamento	lr = 0.001	15	0.20 / 0.21
KNN - ResNet com balanceamento	k = 47	-	0.53 / 0.21

learning rate(lr)

A comparação dos resultados com o classificador Dummy permite descartar a utilização da ResNet como classificador, tanto com o conjunto de dados balanceado como desbalanceado. A utilização das embeddings do modelo no KNN é, entretanto, um método válido. Nesse caso não houve diferença significativa entre o conjunto balanceado e desbalanceado.

A técnica de Early Stopping permite finalizar o treinamento do modelo antecipadamente, armazenando a versão que apresentou o menor loss na validação, o que reduz o tempo total de treinamento. Porém, em todos os testes, o loss de validação foi extremamente elevado quando comparado com o loss de treinamento, além de apresentar uma trajetória crescente ao longo do processo. Esse aumento do loss de validação é característico do processo de overfitting em modelos de deep learning [Zhang et al. 2024], o que demonstra que, mesmo havendo uma queda no loss de treinamento, o modelo ainda é incapaz de generalizar a classificação para imagens não vistas.

Observou-se também que a acurácia do KNN aplicado às embeddings supera, em todos os casos, exceto na CNN com Early Stopping sem balanceamento, o método de classificação direto a partir dos tensores da rede. Essa abordagem, portanto, pode ser considerada eficiente no processo de classificação. No entanto, devido à variabilidade do parâmetro k entre os testes, é fundamental determinar o valor de k que maximize a acurácia.

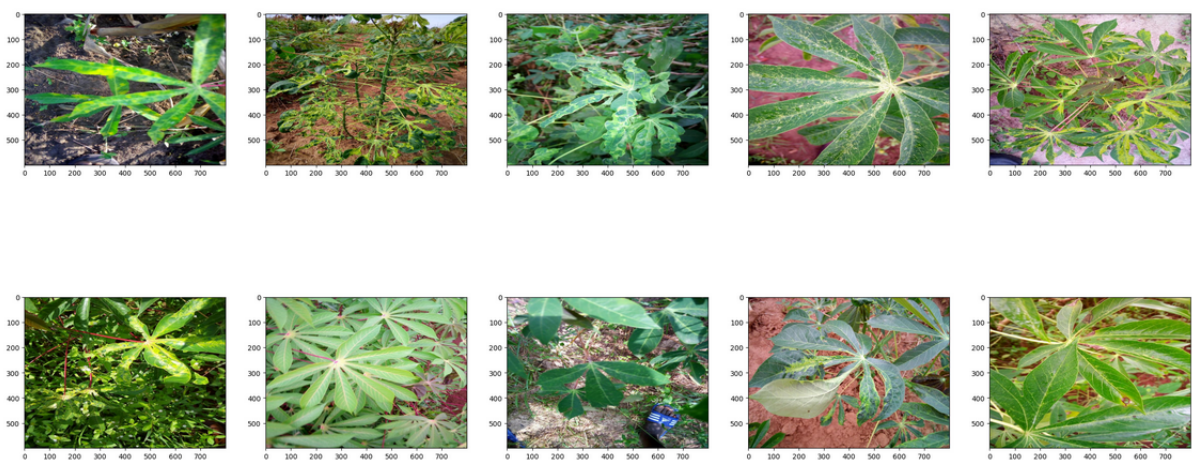
Conclui-se que o modelo que apresentou o melhor desempenho foi a CNN com Early Stopping sem balanceamento. A aplicação do KNN às embeddings geradas pelos modelos também mostrou ser uma técnica eficaz, melhorando significativamente o

<sup>2</sup><https://github.com/pytorch/pytorch/issues/30708>

desempenho da ResNet como classificador. Em quase todos os cenários, com exceção da aplicação direta da CNN com Early Stopping sem balanceamento, o KNN superou a classificação baseada nos tensores do modelo.

## 5. Uso do Conhecimento

Determinar se uma planta é saudável ou possui alguma doença específica mostrou-se uma tarefa desafiadora. Os métodos aplicados para classificação não ultrapassaram o valor de 0.64 no F1-Score ponderado. As imagens apresentadas às redes neurais possuem elementos não pertinentes ao processo de classificação, como pés e mãos das pessoas que capturaram as imagens, além de possuírem ângulos e distância diferentes em relação à câmera como pode ser visto na Figura 3.



**Figura 3. Amostras**

Fica evidente a partir dos resultados e da observação visual das imagens que técnicas de identificação de objetos e de segmentação devem ser aplicadas para eliminação do ruído antes da etapa de treinamento e classificação. Apenas a utilização dos modelos treinados e pré-treinados para essa tarefa de classificação não demonstra ser uma estratégia eficaz.

A utilização do YOLO como técnica de pré-processamento para localizar as folhas ou áreas da planta que têm doenças e excluir elementos irrelevantes é uma possível estratégia.

Outra técnica de pré-processamento para melhorar o resultado dos classificadores é aplicar uma padronização de cor para reduzir variações causadas pela iluminação e pelo ângulo das fotos.

## Referências

Zhang, E., Zhong, R., Munetomo, M., and Wahib, M. (2024). Validation loss landscape exploration with deep q-learning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE.