

# Classificação da Doença da Folha da Mandioca utilizando Redes Neurais Convolucionais

*Cassava Leaf Disease Classification using Convolutional Neural Networks*

**Houston A. Santos<sup>1</sup>**

 [orcid.org/0000-0003-4786-0510](https://orcid.org/0000-0003-4786-0510)

**Lailson B. dos Santos<sup>1</sup>**

 [orcid.org/0000-0002-2811-4443](https://orcid.org/0000-0002-2811-4443)

**Alexandre M. A. Maciel<sup>1</sup>**

 [orcid.org/0000-0003-4348-9291](https://orcid.org/0000-0003-4348-9291)

<sup>1</sup>Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil. E-mail: [has2@ecomp.poli.br](mailto:has2@ecomp.poli.br)

**DOI: 10.25286/repa.v6i5.1754**

Esta obra apresenta Licença Creative Commons Atribuição-Não Comercial 4.0 Internacional.

Como citar este artigo pela NBR 6023/2018: SANTOS, H.A.; SANTOS, L. B. Classificação da Doença da Folha da Mandioca utilizando Redes Neurais Convolucionais. Revista de Engenharia e Pesquisa Aplicada, Recife, v.6, n. 5, p. 47-55, Novembro, 2021.

## RESUMO

Como o segundo maior fornecedor de carboidratos na África, a mandioca é uma cultura de segurança alimentar essencial cultivada por pequenos agricultores porque pode suportar condições adversas. Pelo menos 80% das fazendas familiares na África Subsaariana cultivam essa raiz amilácea, mas as doenças virais são as principais fontes de baixa produção. Os métodos existentes de detecção de doenças exigem que os agricultores solicitem a ajuda de especialistas agrícolas financiados pelo governo para inspecionar visualmente e diagnosticar as plantas. Isso sofre por ser muito trabalhoso, com baixo suprimento e caro. Diante disso, esta pesquisa tem como finalidade a construção de uma máquina preditiva, usando Redes Neurais Profundas (CNN) será possível realizar a identificação do patógeno acometido nas folhas da Mandioca. Para guiar nossa pesquisa foi utilizada a metodologia SEMMA. Os resultados obtidos com o algoritmo construído foram satisfatórios chegando ao resultado de 91% de assertividades nos diagnósticos dos patógenos.

**PALAVRAS-CHAVE:** Rede Neural Convolucional; Mandioca;

## ABSTRACT

As the second largest carbohydrate supplier in Africa, the cassava is an essential food security culture cultivated by small farmers because it can resist adverse conditions. At least 80% of family farms in Sub-Saharan Africa cultivate this starchy root, but the diseases are the main source of low production. The existing disease detection methods require that the farmers request the help of government-funded agricultural experts to visually inspect and diagnose the plants. This suffers for being too laborious, with low supply and expensive. That said, this research aims to build a predictive machine, using Deep Neural Networks (CNN) it'll be possible to identify the pathogen affected in the Cassava leaves. To guide our research the SEMMA methodology was used. The results obtained with the built algorithm were satisfactory, reaching the result of 91% of assertiveness in diagnosing pathogens.

**KEY-WORDS:** Convolutional Neural Network; Cassava;

## 1 INTRODUÇÃO

A mandioca (*Manihot esculenta Crantz*) é uma planta *dicotiledônea*, arbustiva, pertencente à família *Euphorbiaceae*. O produto extraído e mais comercializado é o amido, uma reserva de energia do vegetal que se acumula nas raízes. Além de alimento o amido é usado na indústria têxtil, papelaria e farmacêutica, onde serve de matéria-prima para engomar as fibras e participa da dureza do papel, usada na produção de xarope de glucose e Maltodextrinas e na siderurgia no processo de flotação de ferro [1].

Segundo Francisco [2], a mandioca é cultivada nas regiões tropicais e subtropicais da África, Ásia e América Latina sendo ela a principal fonte de alimentação em diversos países em desenvolvimento. No continente africano a Nigéria é a maior produtora com 55 milhões de toneladas, isso representa 37% da produção de toda a África e 20% da produção mundial. Em segundo lugar vem Gana que vem investindo na produção de alimentos e com o tempo diminuindo os problemas de alimentação.

Nesse contexto, a Embrapa descreve [3] o consumo per capita mundial de mandioca e derivados, em 1996, foi de 17,40 kg por habitante ao ano, no Brasil apresentou foi de 50,60 kg por habitante ao ano. Na África a República Democrática do Congo, República do Congo e Gana apresentaram, respectivamente, valores de 333,2, 281,1 e 247,2 kg por habitante ao ano. Sendo um dos alimentos básicos mais produzidos, a mandioca é muito importante no sub Saahara da África, com uma quantidade de 54% do total da produção mundial (FAO, 2004) [4]. Acredita-se que ficará cada vez mais importante para a população humana e vai continuar avançando em novas áreas (COCKCROFT, 2004).

No entanto, existem pragas que fazem as produções diminuírem bastante como, ácaro da mandioca (*CGM; Cassava Green Mottle*) que evoluiu com a cultura do alimento [5], bacteriose (*CBB; Cassava Bacterial Blight*), o agente que causa essa doença é a bactéria *Xanthomonas axonopodis* pv. *Manihotis* Bondar (Vauterin) (Xam), importante patógeno em campos de mandioca em todo o mundo [6], vírus listrado castanho da mandioca (*CBSD; Cassava Brown Streak Disease*) predomina nas áreas costeiras do leste e sudoeste da África e o mosaico da mandioca (*CMD; Cassava Mosaic Disease*) em todas as regiões são as duas das mais importantes doenças que diminuem a produção da

cultura (CALVERT e THREH, 2002). Muitas dessas doenças causadas por vírus não ocorrem no Brasil, mas são de grande importância se forem introduzidas em nossas condições de clima e solo. O mosaico comum e o mosaico das nervuras são doenças de etiologia viral que tem sido encontrada em todas as áreas de produção de mandioca, em incidência baixa (TAKATSU e FUKUDA, 1990) [4].

Diante dos fatos apresentados o estudo está direcionado na criação de uma máquina preditiva, um classificador de imagens, capaz de identificar os patógenos (*CGM, CBB, CBSD, CMD*) e folhas saudáveis.

Tendo como principal base de dados 21.397 imagens coletados durante uma pesquisa em Uganda, a maioria das imagens foram fotografadas por agricultores de seus próprios jardins, (os agricultores não têm acesso a alta tecnologia e usam celulares com câmeras de baixa resolução e com pouca largura de banda, sendo esse um desafio para diagnosticar a doença) e anotadas por especialistas do *National Crops Resources Research Institute (NaCRRI)* em colaboração com o laboratório de IA da *Makerere University, Kampala*. O **Makerere Artificial Intelligence (AI) Lab** é um grupo de pesquisa em IA e ciência de dados baseado na *Makerere University* em Uganda. O laboratório é especializado na aplicação de inteligência artificial e ciência de dados, incluindo, por exemplo, métodos de aprendizado de máquina, visão computacional e análise preditiva para problemas no mundo em desenvolvimento.

A metodologia *SEMMA* foi utilizada por se adaptar melhor ao projeto de Redes Neurais. Durante toda pesquisa e elaboração faremos uso da linguagem de programação Python e frameworks como o *Keras Tensorflow, Scikit-learn, Pandas, Numpy* entre outros. Como principal técnica de aprendizado supervisionado para Redes Neurais irá adotar o aprendizado por transferência (*Transfer Learning*).

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, abordamos de forma breve e objetiva o aprendizado por transferência (*Transfer Learning*), e a metodologia *SEMMA*.

### 2.1 Aprendizagem por Transferência

Segundo Torrey e Shavlik [7], objetivo da transferência de aprendizagem é melhorar a aprendizagem na tarefa alvo, aproveitando o conhecimento da tarefa de origem. Existem três medidas comuns pelas quais a transferência pode melhorar a aprendizagem. O primeiro é o desempenho inicial alcançável na tarefa alvo usando apenas o conhecimento transferido, antes que qualquer aprendizado adicional seja feito, em comparação com o desempenho inicial de um agente ignorante. O segundo é a quantidade de tempo que leva para aprender totalmente a tarefa alvo, dado o conhecimento transferido, em comparação com a quantidade de tempo para aprendê-la do zero. O terceiro é o nível de desempenho final alcançável na tarefa alvo em comparação com o nível final sem transferência.

## 2.2 SEMMA

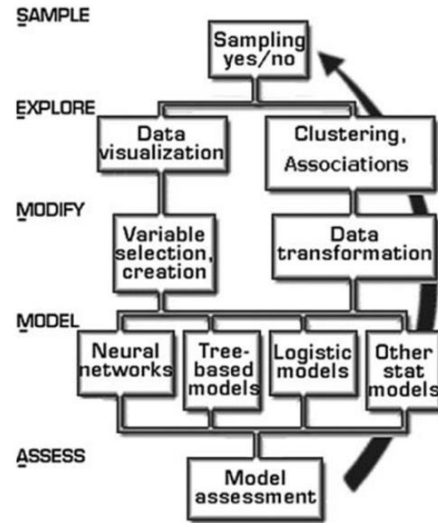
SEMMA é um acrônimo para as palavras *Sample, Explore, Modify, Model e Assess*. Desenvolvida pela empresa *SAS Enterprise Miner* como uma organização lógica para o uso de ferramentas de mineração de dados [8].

*"SEMMA is not a data mining methodology but rather a logical organization of the functional tool set of SAS Enterprise Miner for carrying out the core tasks of data mining. Enterprise Miner can be used as part of any iterative data mining methodology adopted by the client."* SAS (2006)

Nogueira [9] descreveu o SEMMA como sendo um processo fácil de perceber, permitindo organizar e adequar o desenvolvimento e manutenção de projetos de *data mining*. Apresenta uma estrutura para concepção, criação e evolução, ajudando a apresentar uma solução. Partindo de uma amostra representativa dos dados, o SEMMA tem como objetivo facilitar a aplicação de técnicas de visualização e exploração estatística, selecionar e transformar as variáveis mais significativas, modelar as variáveis de forma a prever resultados, e por fim confirmar a precisão do modelo. A Figura 1 ilustra os processos da metodologia SEMMA.

Em sua pesquisa, Ana Azevedo [10] mostra as cinco etapas para o processo de desenvolvimento SEMMA de acordo com *SAS Institute*, conforme apresenta o Quadro 1.

Figura 1: Ilustração do processo SEMMA



Fonte: Nogueira (2014).

Quadro 1: Etapas de desenvolvimento SEMMA.

FASES	DEFINIÇÃO
Sample	Amostrar os dados extraíndo uma parte de um grande conjunto de dados grande o suficiente para conter as informações significativas, mas pequeno o suficiente para manipular rapidamente.
Explore	Exploração dos dados em busca de tendências e anomalias imprevistas, a fim de obter compreensão e ideias.
Modify	Modificação dos dados por meio da criação, seleção e transformação das variáveis para enfatizar o processo de seleção do modelo.
Model	Modelar os dados, permitindo que o software busque automaticamente uma combinação de dados que preveja com segurança o resultado desejado.
Assess	Avaliar os dados avaliando a utilidade e confiabilidade dos achados do processo de DM e estimar o seu desempenho.

Fonte: Os autores.

## 3 MATERIAIS E MÉTODOS

As subseções a seguir aborda a construção da solução de *Machine Learning* proposta em pesquisa, dentro da metodologia utilizada.

### 3.1 DADOS DA AMOSTRA

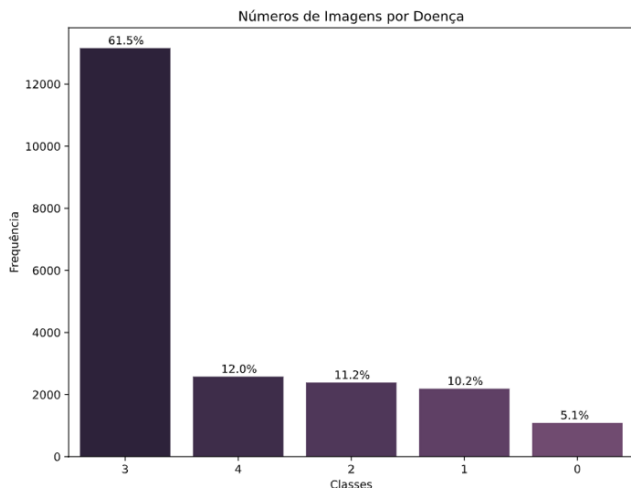
O banco de imagens utilizado é da *National Crops Resources Research Institute (NaCRRRI)* em colaboração com o laboratório de IA da *Makerere*

University, Kampala. Nele temos um total de 21397 imagens divididas da seguinte forma:

- Classe 0 - CBB
- Classe 1 - CBSD
- Classe 2 - CGM
- Classe 3 - CMD
- Classe 4 - Folha Saudável

Podemos observar na Figura 2 a frequência e como as imagens estão relativamente distribuídas.

**Figura 2:** Frequência das imagens.



Fonte: Os autores.

### 3.2 PREPARAÇÃO DOS DADOS

Como podemos observar na Figura 2, a classe 3 apresenta uma quantidade considerável em relação as demais, o desbalanceamento faz o modelo aprender mais sobre a classe predominante em relação as outras, esse tipo de problema é conhecido como classe rara. Amaral [11] propôs uma solução para o problema da classe rara através da estratificação. Para realizar esse processo dados foram separados utilizando a técnica *HoldOut* de forma estratificada pela classe, mantendo suas proporções em cada separação; 70% (14977 para treino), 30% (6420 para teste), das imagens de treino 10% (642 para validação). Os dados de validação são usados ao final, com a intensão de simular um ambiente real de produção.

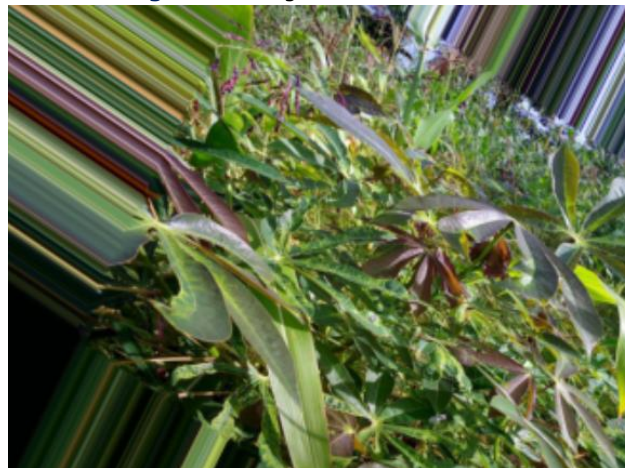
### 3.3 TRATAMENTO DOS DADOS

As imagens não estão no formato adequado para utilizarmos em uma rede neural, usaremos a classe **ImageDataGenerator** com o método **flow\_from\_dataframe** do keras para percorrer o arquivo cvs e diretório de dados, carregando os

dados de imagem e retornar a entrada (matrizes de pixels) e a saída (número inteiro de classe). Com isso os conjuntos de imagens de treino, teste e validação serão transformados em uma matriz de tamanho  $n$ , onde  $n$  é o número de imagens do conjunto observado ( $n, 456, 456, 3$ ) as imagens originais têm uma resolução de  $600 \times 800$  pixels. A classe **ImageDataGenerator** ainda oferece o recurso de *Data Augmentation*.

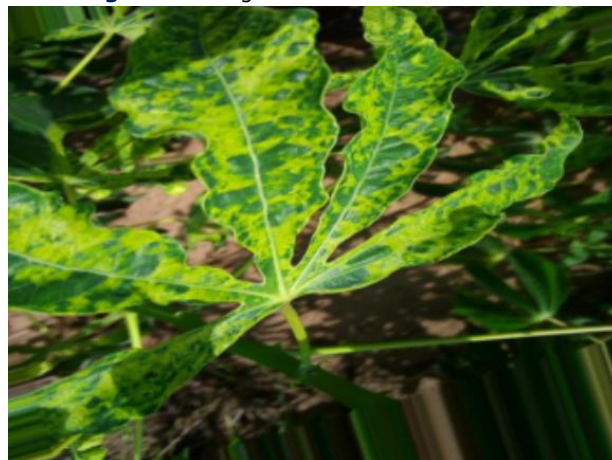
É uma técnica para gerar novos exemplares de dados de treinamento a fim de aumentar a generalidade do modelo. Técnica de aumento de dados como corte, preenchimento, rotação e redimensionamento são comumente usadas para treinar grandes redes neurais [12]. As figuras 3, 4 e 5 demonstram as imagens com a aplicação do *Data Augmentation*.

**Figura 3:** Imagem folha saudável.



Fonte: Os autores.

**Figura 4:** Imagem folha doente com CMD.



Fonte: Os autores.

**Figura 5:** Imagem folha doente com CBSD.





Fonte: Os autores.

### 3.4 CONSTRUÇÃO DO MODELO

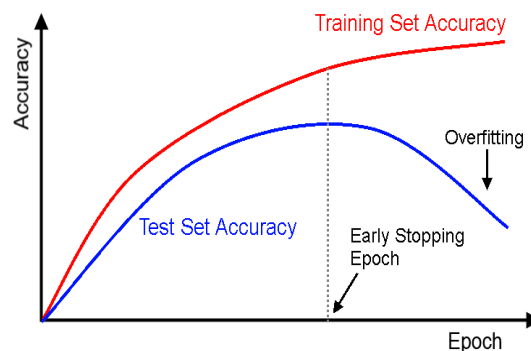
#### 3.4.1 Hiperparâmetros

Ao treinar redes neurais, várias decisões precisam ser tomadas em relação às configurações usadas, a fim de obter um bom desempenho. Uma boa opção de ajuste são os hiperparâmetros.

Um *callback* de chamada é um conjunto de funções de hiperparâmetros a serem aplicadas em determinados estágios do treinamento. Com isso podemos obter uma visão dos estados e estatísticas do modelo durante o treinamento, três *call-backs* foram utilizados, são eles:

- **ModelCheckpoint** – Ao fim de cada epoch a função vai verificar o valor da *val\_loss*, e comparar, salvando o modelo e seus pesos sempre que obtiver o menor valor para *val\_loss*.
- **ReduceLROnPlateau** – Checa o valor da *val\_loss* ao final de cada epoch, se a mesma por duas epochs estagnar, irá reduzir a taxa de aprendizagem (*learning rate*).
- **EarlyStopping** – Irá parar o treinamento se o valor da *val\_loss* parar de diminuir por três epochs, entendendo que o modelo chegou em seu melhor ajuste, evitando assim o *overfitting*, na Figura 6 podemos verificar e entender melhor o funcionamento do *EarlyStopping*.

Figura 6: Funcionamento da *EarlyStopping*.



Fonte: Os autores.

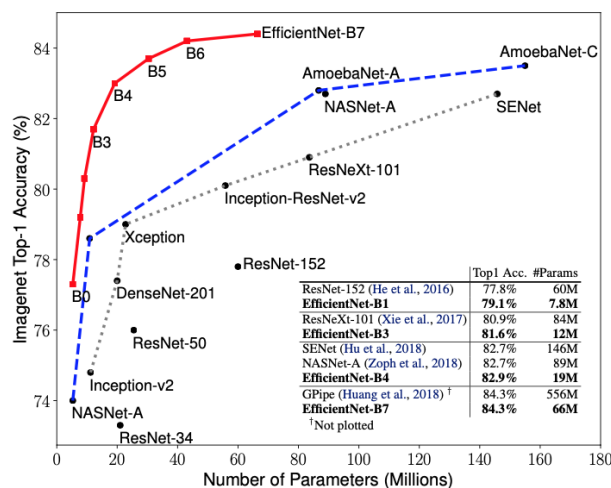
#### 3.4.2 Camada Convolutacional

Para realização do aprendizado por transferência (*transfer learning*) na camada convolutacional da *CNN* em pesquisa foi utilizada a *EfficientNet-B3*, a *EfficientNet* que está entre os modelos mais eficientes quando se trata de classificar imagens, fornece uma família de modelos (B0 a B7) e tem se mostrado uma das melhores *ConvNets* em diversas pesquisas utilizando o conjunto de dados *ImageNet*.

O *EfficientNet-B7* atinge um estado da arte de 84,3% da precisão top-1, mas sendo 8,4x menor e 6,1x mais rápido do que o *GPipe*.

O *EfficientNet-B1* é 7,6x menor e 5,7x mais rápido do que o *ResNet-152* [13]. Na Figura 7 podemos ver um comparativo da *EfficientNet* com diversas *ConvNets*.

Figura 7: Comparativo *EfficientNet* vs *ConvNets*.



Fonte: Os autores.

Após a criação das camadas subsequentes *GlobalAveragePooling2D*, *Flatten*, *Dense*, *Dropout*,

Dense e compilação da rede neural, podemos ver seu resumo na Figura 8.

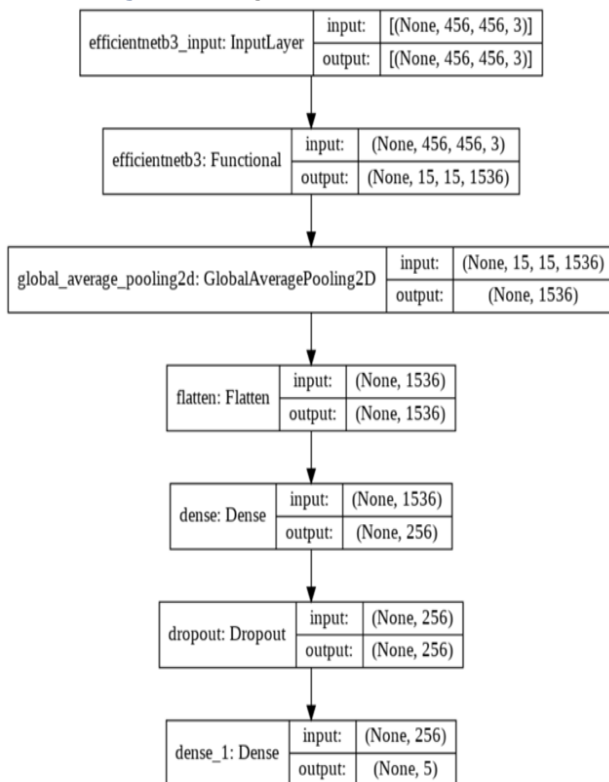
**Figura 8:** Resumo CNN.

Layer (type)	Output Shape	Param #
efficientnetb3 (Functional)	(None, 15, 15, 1536)	10783535
global_average_pooling2d_1 ( (None, 1536)		0
flatten_1 (Flatten)	(None, 1536)	0
dense_2 (Dense)	(None, 256)	393472
dropout_1 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 5)	1285
Total params: 11,178,292		
Trainable params: 11,090,989		
Non-trainable params: 87,303		

**Fonte:** Os autores.

Na Figura 8 foi possível verificar o total de parâmetros de cada camada, e *shape* das mesmas, porém para termos uma melhor ideia do input e output dessas camadas e o fluxo de funcionamento da rede neural, a Figura 9 ilustra um diagrama de fluxo da rede neural.

**Figura 9:** Diagrama de Fluxo da CNN.



**Fonte:** Os autores.

## 4 RESULTADOS

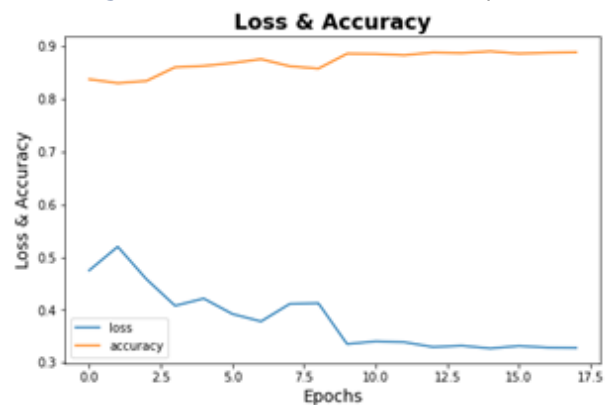
Nas subseções a seguir apresentamos os resultados de duas abordagens na execução da rede neural. Sendo elas *HoldOut* e *Cross-Validation*.

### 4.1 HOLDOUT

Consiste em uma separação simples do conjunto de dados entre treino e teste, onde o conjunto de dados pode ser separado em partes iguais ou não, uma proporção muito comum é 2/3 para treino e 1/3 para teste. Após o modelo ser criado e executado os testes são aplicados e a predição calculada [14]. Para o experimento em pesquisa foi levado em consideração as proporções das classes.

Para o experimento utilizou-se 70% para treino e 30% para teste. Na Figura 10 temos um gráfico de linhas onde podemos verificar e comparar a evolução da *val\_loss* e *val\_categorical\_accuracy*.

**Figura 10:** HoldOut - Loss & Accuracy.



**Fonte:** Os autores.

A Figura 11 traz um comparativo de desempenho da rede neural entre os valores da *loss* (treino), *categorical\_accuracy* (treino) e *val\_loss* (teste) e *val\_categorical\_accuracy* (teste).

**Figura 11:** Performance Loss & Accuracy.



**Fonte:** Os autores.

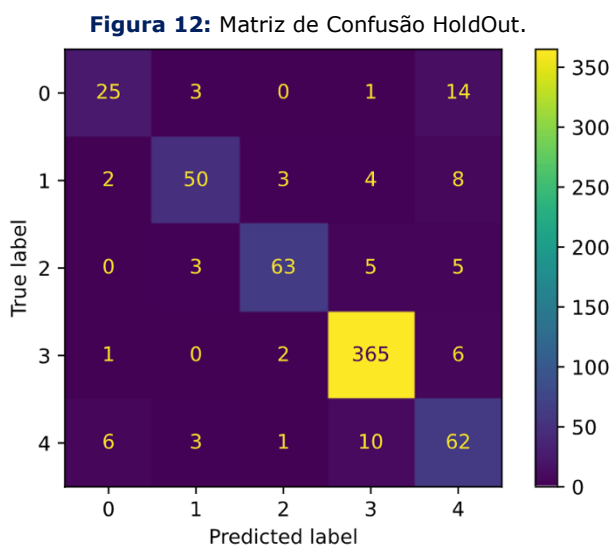
Neste primeiro experimento foram obtido os seguintes resultados conforme Tabela 1.

**Tabela 1:** Resultado – HoldOut

	LOSS	ACCURACY
TREINO	0.2898	0.9016
TESTE	0.3280	0.8888
VALIDAÇÃO	0.3678	0.8863

Fonte: Os autores.

O conjunto de validação possui 642 imagens nunca vistas antes pelo modelo, na Figura 12 temos uma matriz de confusão, que ilustra o resultado de validação.



Fonte: Os autores.

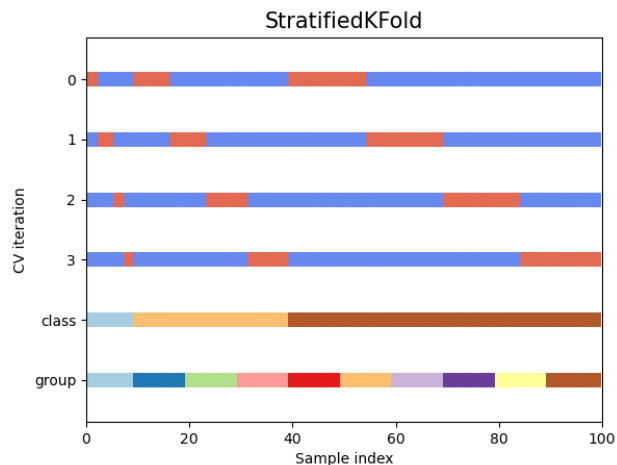
## 4.2 CROSS VALIDATION

Neste segundo experimento, após aplicação do *HoldOut*, com os dois sets, treino e teste, foi executado a validação cruzada mais conhecida como cross-validation. A validação é um método estatístico que consiste em comprar algoritmos por meio de uma divisão de dados separados em dois segmentos, treino e teste essa separação dos dados dar-se o nome de *folds* (partes iguais), ao qual irá treinar e testar o modelo n (*folds*) vezes, com ajuda do scikit-learn usando *StratifiedKFold* podemos realizar esta separação de forma proporcional as classes, na Figura 13 temos uma ilustração de ficou nosso set [15].

Na execução da validação cruzada foram utilizados 5 *folds* de forma estratificada, ou seja, mantendo a proporção das classes observadas. Neste método o teste não é executado ao final de

cada epoch, como acontece quando utilizamos o *HoldOut*, e sim ao final do treinamento de cada *fold*, na Tabela 2 podemos observar os valores obtidos dos testes.

**Figura 13:** Cross-Validation – StratifiedKFold.



Fonte: Os autores.

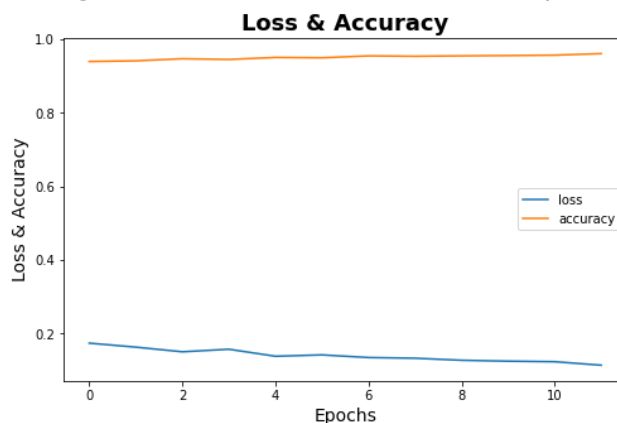
**Tabela 2:** Resultado – Validação Cruzada.

FOLD	LOSS	ACCURACY
1	0.3905	0.8733
2	0.3265	0.8950
3	0.2629	0.9114
4	0.1869	0.9340
5	0.1519	0.9446
Média	0.2637	0.9117
Média D. Padrão	0.0876	0.0258

Fonte: Os autores.

Podemos verificar na Figura 14 o comportamento da rede neural, e observar uma menor variação quando comparada ao primeiro experimento.

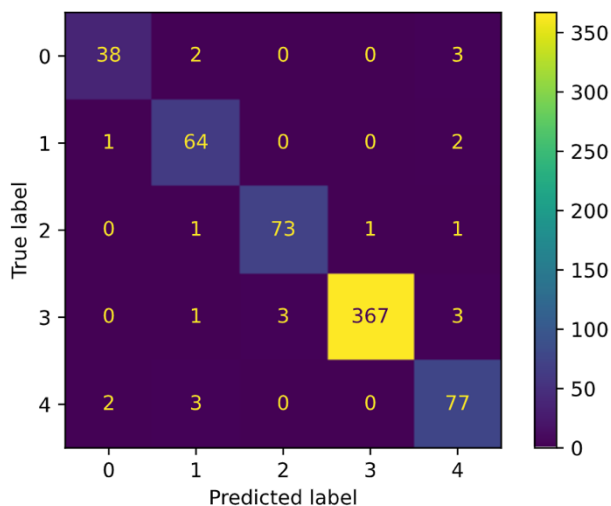
**Figura 14:** Cross-Validation - Loss & Accuracy.



Fonte: Os autores.

Por fim, o segundo experimento foi submetido ao conjunto de dados de validação, o mesmo utilizado para o primeiro experimento usando *Holdout*, obtivemos os seguintes resultados; loss: 0.0969 e accuracy: 0.9642. A Figura 15 traz a matriz de confusão.

**Figura 15:** Matriz de Confusão Cross-Validation.



Fonte: Os autores.

## 5 CONCLUSÕES

Diante dos resultados aqui apresentados ao longo da pesquisa, onde ambos os experimentos foram submetidos ao mesmo conjunto de dados de validação com 642 imagens, vemos que o primeiro experimento usando *HoudOut* teve uma boa performance, mas não o suficiente para o nosso objetivo que era mais de 90% ou mais de acurácia, ficando apenas com 89% e acertando 565 das imagens apresentadas.

Sendo assim, o segundo experimento foi submetido a validação cruzada que trouxe o resultado desejado, chegando a acertar 619 das 642 imagens, isso representa 96% de assertividade.

Realizada a criação da máquina preditiva, a fim de classificar doenças nas folhas da mandioca, foi criado uma página e disponibilizada publicamente que pode ser acessada através do link [https://share.streamlit.io/houstonsantos/appcassa\\_valeafdisese/src/streamlit\\_app.py](https://share.streamlit.io/houstonsantos/appcassa_valeafdisese/src/streamlit_app.py). Essa página consiste em realizar a classificação do patógeno, encontrado em uma imagem da folha da mandioca e apresentar um pequeno resumo sobre a doença presente na folha.

## REFERÊNCIAS

[1] MORAIS, M. S.; MEDEIROS, E. V.; MOREIRA, K. A.; CAVALCANTI, M. S.; OLIVEIRA, N. T. Epidemiologia das doenças da parte aérea da Mandioca no Município de Alagoa Nova, Paraíba. *SciELO Brasil*, v.40, n.3, p.264-269, 2014.

[2] JUNIOR, F. A. G. Produtividade de Variedades de Mandioca em Diferentes Arranjos de Plantio, Épocas de Colheita, Fisiologia do Estresse e Déficit Hídrico. Bahia, UFRB 2018.

[3] A Cultura da Mandioca. Sistema de Produção Embrapa. Disponível em: <<https://www.bibliotecaagptea.org.br/agricultura-novo/culturas-anuais/culturas-anuais-livros/>> Acesso em: 15 abr. 2021.

[4] ARALDI, R.; SILVA, I. P. F.; TANAKA, A. A.; GIOTTO, M.; SILVA JUNIOR, J. F. Doenças Virais na Cultura da Mandioca. *FAEF Revista Científica Eletrônica de Agronomia*, n. 20 de 2011.

[5] Indução de Resistência em Manihot Esculenta Crantz ao Ataque de *Vatiga Illudens*. Disponível em: <<https://cointer.institutoidv.org/inscricao/pdvagro/uploadsAnais/INDU%C3%87%C3%83O-DE-RESIST%C3%8ANCIA-EM-Manihot-esculenta-Crantz-AO-ATAQUE-DE-Vatiga-illudens.pdf>> Acesso em: 15 abr. 2021.

[6] OLIVEIRA, S. A. S.; SILVA, M. A.; RANGEL, M. A. S.; SANTOS, V. S.; RINGENBERG, R.; OLIVEIRA, E, J. Metodologia para Avaliação da Resistência da Mandioca à Bacteriose, Antracnose e Superalongamento. Disponível em: <<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/146099/1/boletim-78.pdf>> Acesso em: 15 abr. 2021.

[7] TORREY, L.; SHAVLIK, J. (2009), *Transfer Learning*. Madison, WI: University of Wisconsin.

[8] MOURA, Karina. *Data Science: Um estudo dos métodos no mercado e na academia*, Porto Alegre: Universidade Federal do Rio Grande do Sul, 2018. Página 31.

[9] NOGUEIRA, D. *Agile Data Mining: Uma metodologia ágil para o desenvolvimento de projetos de data mining*. Porto, Portugal: Universidade do Porto, 2014, p.6-8.



**[10]** AZEVEDO, A. I. R. L.; SANTOS, M. F. KDD, SEMMA and CRISP-DM: a parallel overview. IADIS European Conference on Data Mining, 2008, IADIS. p.182–185.

**[11]** AMARAL, F. Introdução à Ciência de Dados: Mineração de Dados e Big Data: Rio de Janeiro: Alta Books; 2016.

**[12]** Agrupando conceitos e classificando imagens com Deep Learning.  
Disponível em:  
<<https://estevestoni.medium.com/agrupando-conceitos-e-classificando-imagens-com-deep-learning-5b2674f99539>> Acesso em: 29 abr. 2021.

**[13]** TAN, M.; LE, V. Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020. Disponível em <<https://arxiv.org/pdf/1905.11946.pdf>>. Acesso 18 abril 2021.

**[14]** KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International joint Conference on artificial intelligence. [S.l.: s.n.], 1995. v. 14, p. 1137–1145.

**[15]** REFAEILZADEH, P.; TANG, L.; LIU, H. Cross-Validation. Arizona State University, 2008. Disponível em: <<https://web.archive.org/web/20110905044421/http://www.public.asu.edu/~ltang9/papers/ency-cross-validation.pdf>>. Acesso em: 19 abril 2021.