

CLASSIFICAÇÃO MULTICLASSE DE DOENÇAS E PRAGAS EM FOLHAS DE CAFÉ USANDO APRENDIZADO PROFUNDO

Clécio Elias Silva e Silva

Centro de Ciências Exatas e Tecnológicas
Universidade Federal do Acre

Progresso da Pesquisa, 2024

Agenda

- ① Introdução
- ② Revisão Bibliográfica
- ③ Objetivo
- ④ Metodologia
- ⑤ Referência
- ⑥ Avanços 10/11
- ⑦ Próximos Passos
- ⑧ Próximos Passos

Introdução

Pesquisa anterior e artigo publicado na SBC.

PavicNet-MC: Um modelo de classificação *multilabel* aplicado em ultrassonografia pulmonar

Clécio Elias Silva e Silva¹, Salomão Mafalda Machado¹, Ana Beatriz Alvarez¹,
Roger Fredy Larico Chavez¹

¹Centro De Ciências Exatas e Técnologicas – Universidade Federal do Acre (UFAC)
Rio Branco – AC – Brasil

{clecio.silva, salomao.machado}@sou.ufac.br, {ana.alvarez, roger.fredy}@ufac.br

Abstract. In recent years, the consequences of Covid-19 and other lung diseases have caused an increase in demand for health services, early and accurate diagnosis of these diseases is essential for the recovery of patients. This article proposes a multilabel classification model, called PavicNet-MC. This model was developed with the aim of identifying five features visible on lung ultrasound. The proposed model obtained an accuracy of 99% in the classification of the five characteristics. Results show that the proposed model is highly effective in detecting and monitoring the visible features that correlate with lung diseases, and has a relatively low complexity compared to other architectures found in the literature.

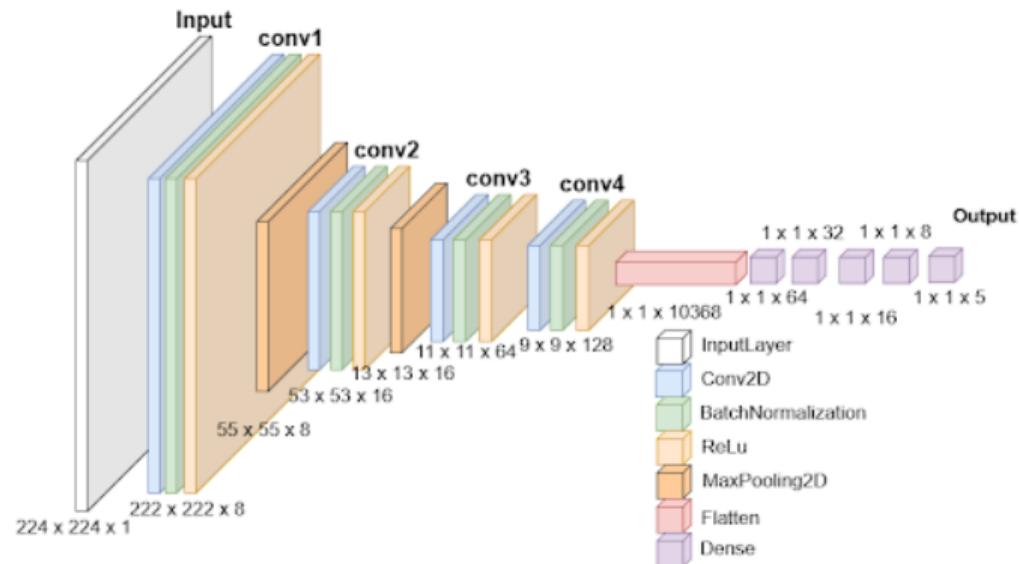
Resumo. Nos últimos anos as consequências da Covid-19 e outras doenças pulmonares vem causando um aumento na demanda pelos serviços de saúde, o diagnóstico precoce e preciso dessas doenças é essencial para a recuperação dos pacientes. Este artigo propõe um modelo de classificação multirrotulo, denominado PavicNet-MC. Este modelo foi desenvolvido com a motivação de identificar cinco características visíveis em ultrassonografia pulmonar. O modelo proposto obteve uma precisão de 99% na classificação das cinco características



Introdução

Arquitetura PavicNet-MC:

- Eficaz na classificação multilabel
- Tem um baixo custo computacional (Poucos parâmetros comparado a outras arquiteturas)
- Pode ser facilmente adaptado para outros problemas



Doenças e pragas nas folhas do Café Brasileiro mesmo com safra podada (Julho de 2023).



Lavouras de café com safra zero também devem receber controle de doenças

POR JOSÉ BRAZ MATIELLO
FOLHA PROCAFÉ
EM 11/07/2023
2 MIN DE LEITURA

1

[Compartilhar](#) [Curte](#) 1

[f](#) [X](#) [in](#) [WhatsApp](#)

[Início](#) > [Colunas](#) > [Folha Procafé](#) > Lavouras de café com safra zero também devem receber controle de doenças

O controle de doenças no cafeiro também é importante em lavouras que se encontram com safra zerada por poda.

1

¹ fonte: <https://www.cafepoint.com.br/colunas/folha-procafe-jose-braz-matiello/>

Revisão Bibliográfica

Pequena revisão bibliográfica sobre conceitos abordados neste trabalho:

- ① Classificação Binária
- ② Classificação Multiclasses

a) Classificação Binária



Saudável

Doente

b) Classificação Multiclasses



Saudável

Phoma

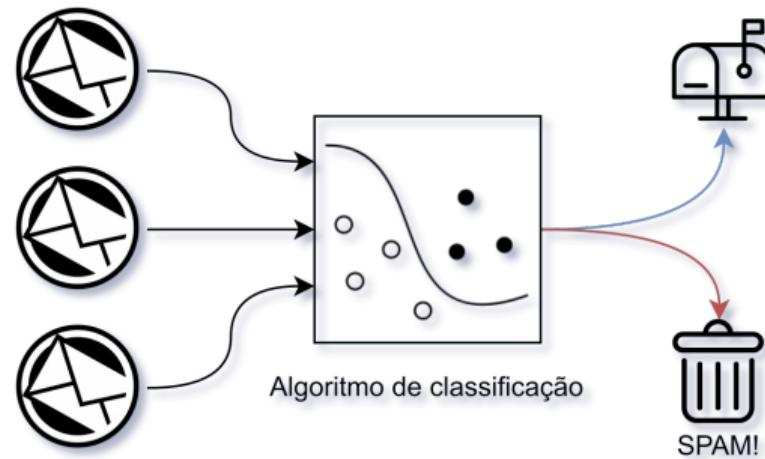
Rust

Miner

Cercospora

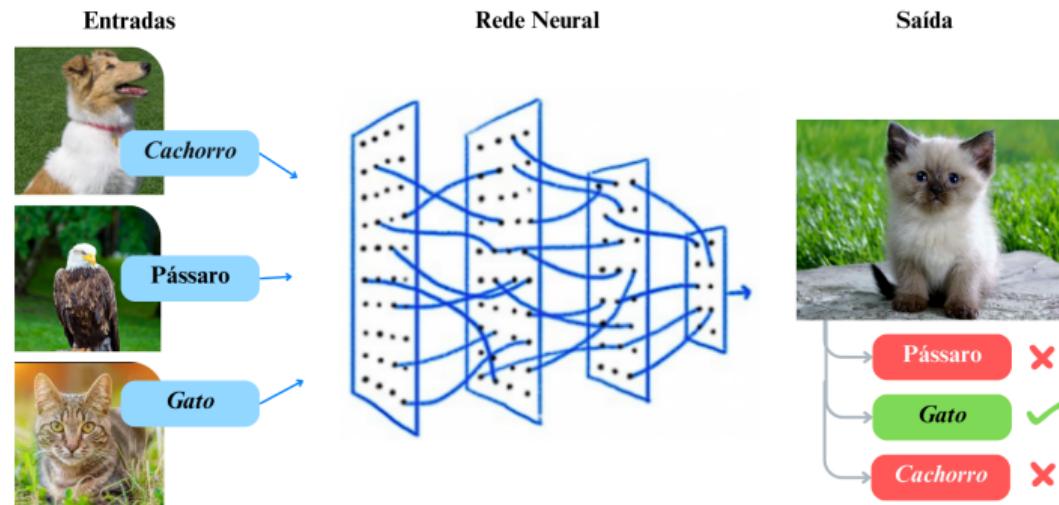
Classificação

Classificação Binária ou simplesmente Classificação tem o objetivo de atribuir uma instância a uma de duas classes mutuamente exclusivas.



Classificação Multiclasse

A classificação multilabel lida com problemas em que uma instância pode pertencer a várias classes simultaneamente.



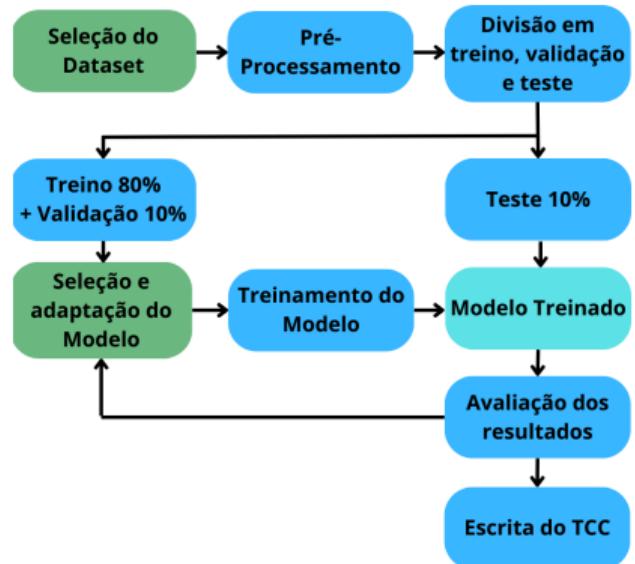
Objetivo

- Classificação automática de doenças e pragas em folhas de café propondo uma arquitetura de classificação eficaz e de baixo custo computacional.



Objetivo

- Realizar o pré-processamento dos datasets JMuBEN, BRACOL, RoCoLe e DiseasesInCoffeeLeaf.
- Adaptar a arquitetura PavicNet-MC para otimizar sua eficácia na classificação de doenças e pragas em folhas de café.
- Ajustar e configurar os modelos ShuffleNet, ResNet50, InceptionResNetV2, MobileNet V2 e DenseNet169.
- Treinar os modelos e avalia-los para classificar em 5 classes: Phoma, Rust, Miner, Cercospora e Healthy.
- Comparar o desempenho dos modelos adaptados com o modelo PavicNet-MC.



Dataset (JMuBEN)



Healthy



Rust



Miner



Phoma



Cercospora

Dataset (BRACOL)

Healthy, Cercospora, Miner, Rust and Phoma.



Dataset (RoCole e Disease and pest in coffee leaves)

Rust and Red Spider Mite.



Miner and Rust



Metodologia

- Hardware e Software:
 - Python (Keras e Tensorflow)
 - Google Colab
- Métricas de Avaliação:
 - Matriz de Confusão
 - Acurácia
 - Precisão
 - Área sobre a curva ROC
- Dataset (JMuBEN, BRACOL, RoCoLe e DPCL)



Web-based CNN Application for Arabica Coffee Leaf Disease Prediction in Smart Agriculture

Accredited Ranking SINTA 2

Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: <http://jurnal.iaii.or.id>



JURNAL RESTI

(**Rekayasa Sistem dan Teknologi Informasi**)

Vol. 7 No. 1 (2023) 72 - 79

ISSN Media Electronic: 2580-0760

Web-based CNN Application for Arabica Coffee Leaf Disease Prediction in Smart Agriculture

Yazid Aufar¹, Muhammad Helmy Abdillah², Jiki Romadoni³

^{1,3}Informatics Engineering, Politeknik Hasnur

²Cultivation of Plantation Plants, Politeknik Hasnur

¹yazid.aufar.ya@gmail.com, ²abdillah.helmy21@gmail.com, ³jiki.romadoni@gmail.com

Abstract

In the agriculture industry, plant diseases provide difficulty, particularly for Arabica coffee production. A first step in eliminating and treating infections to avoid crop damage is recognizing ailments on Arabica coffee leaves. Convolutional neural networks (CNN) are rapidly advancing, making it possible to diagnose Arabica coffee leaf damage without a specialist's help. CNN is aimed to find features adaptively through backpropagation by adding layers including convolutional layers and pooling layers. This study aims to optimize and increase the accuracy of Arabica coffee leaf disease classification utilizing the neural network architectures: ResNet50, InceptionResNetV4, MobileNetV2, and DensNet169. Additionally, this research presents an interactive web platform integrated with the Arabica coffee leaf disease prediction system. Inside this research, 5000 image data points will be divided into five classes—Phoma, Rust, Cescospora, healthy, and Miner—to assess the efficacy of CNN architecture in classifying images of Arabica coffee leaf disease. 80:10:10 is the ratio between training data, validation, and testing. In the testing findings, the InceptionResNetV2 and DensNet169 designs had the highest accuracy, at 100%, followed by the MobileNetV2 architecture at 99% and the ResNet50 architecture at 59%. Even though MobileNetV2 is not more accurate than InceptionResNetV2 and DensNet169, MobileNetV2 is the smallest of the three models. The MobileNetV2 paradigm was

Próximos Passos

Curto Prazo:

Replicar o trabalho do artigo de referência:

- Implementar as 4 arquiteturas restantes
- Iniciar o treinamento e validação dessas arquiteturas

Médio Prazo:

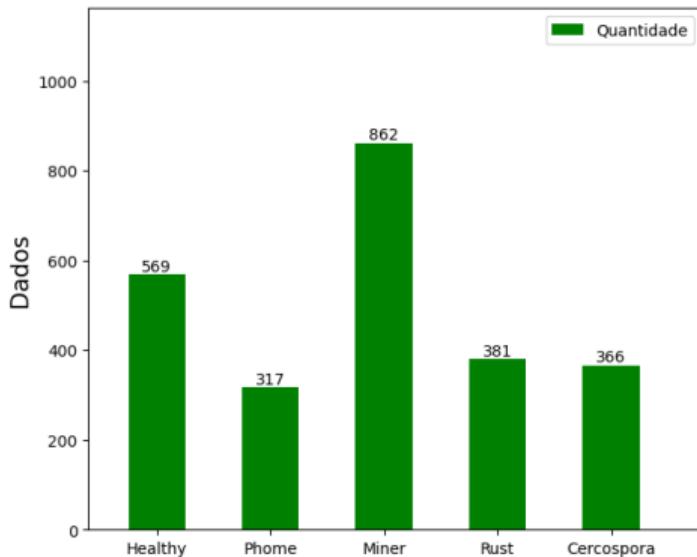
- Dar continuidade a escrita do TCC

Implementado as arquiteturas do artigo de referência:

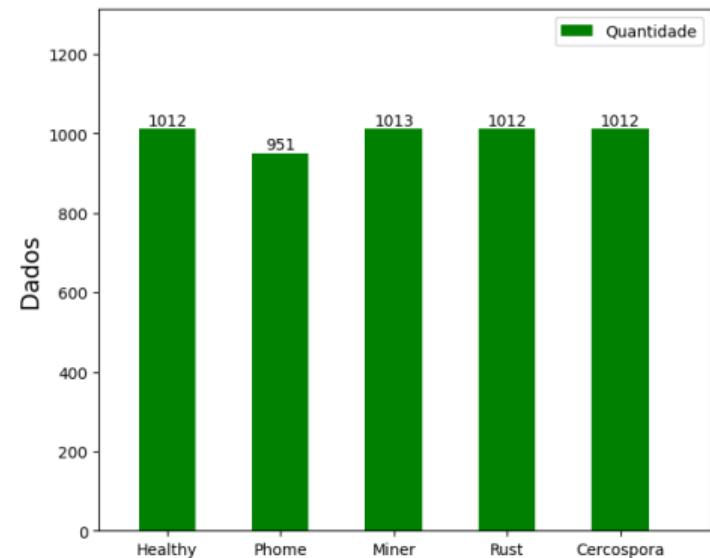
- Resnet50
- InceptionResnetV2
- MobilenetV2
- DenseNet169
- Balanceamento do dataset com base no artigo base

Resultados preliminares

Dataset JMUBEN original



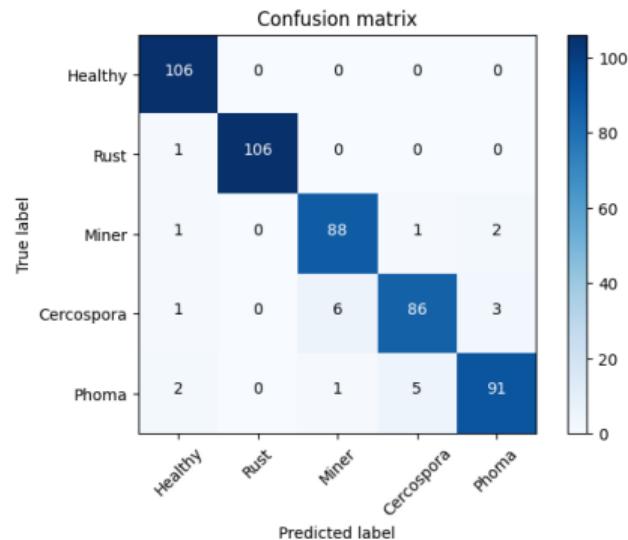
Dataset JMUBEN original com data augmentation e balanceamento



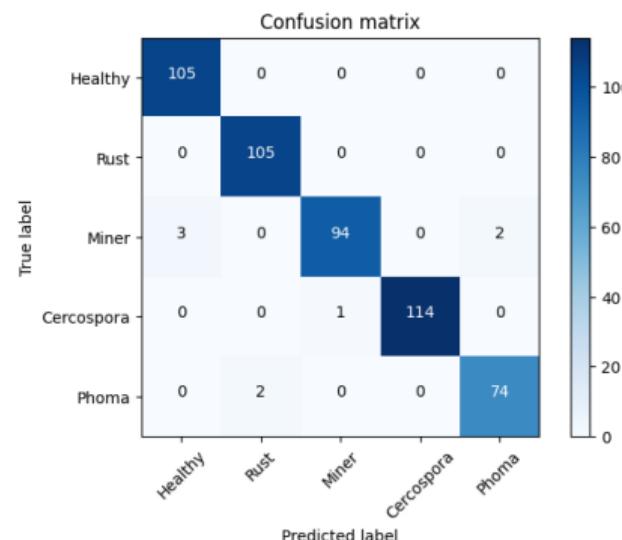
Resultados Preliminares

Resultados preliminares:

Modelo: PavicNet-MC

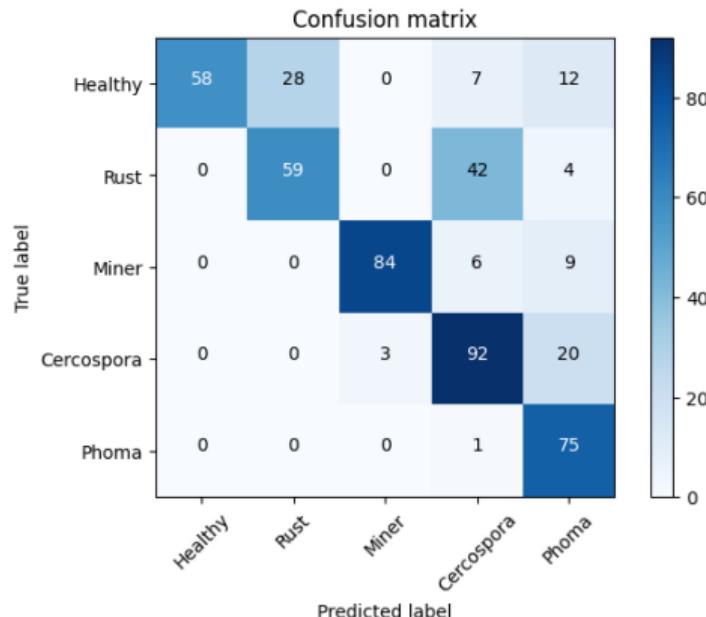


Modelo: InceptionResnetV2



Resultados Preliminares

Modelo: Resnet50



Próximos Passos

Curto Prazo:

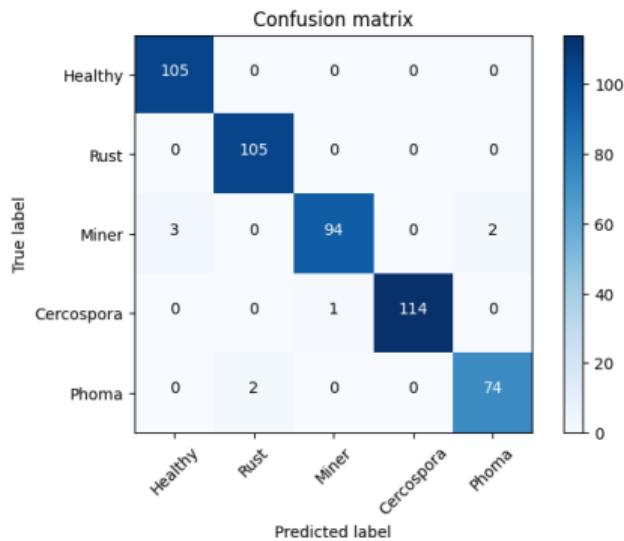
- Terminar o treinamento e validação dessas arquiteturas
- Comparar resultados com o artigo de referência

Médio Prazo:

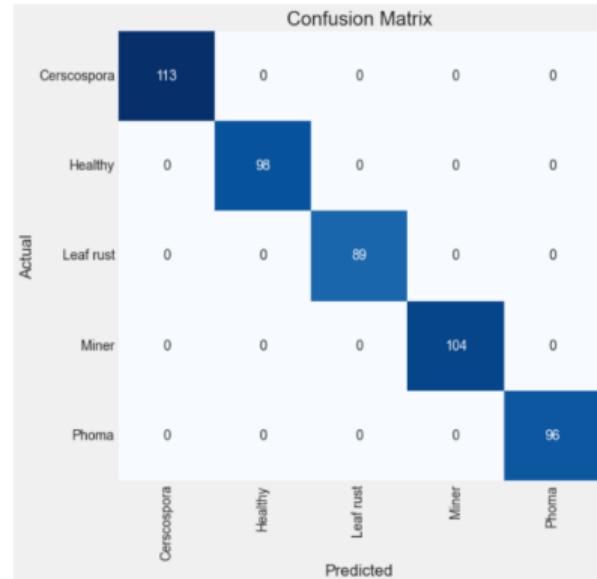
- Dar continuidade a escrita do TCC

Resultados Preliminares

Modelo: InceptionResnetV2 Treinada

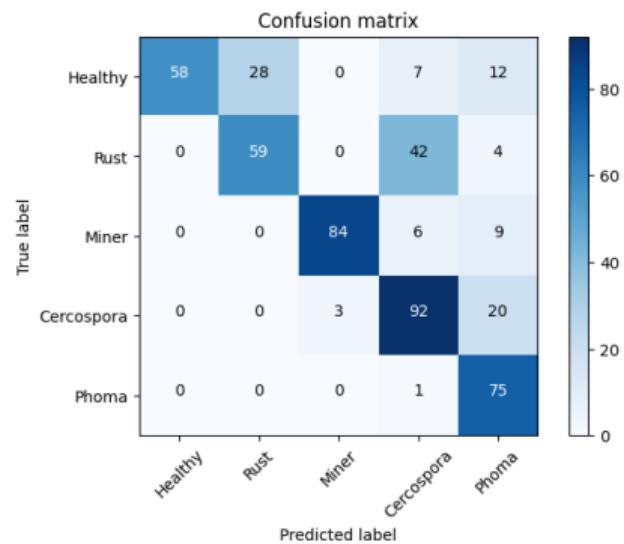


Modelo: InceptionResnetV2 de Referência

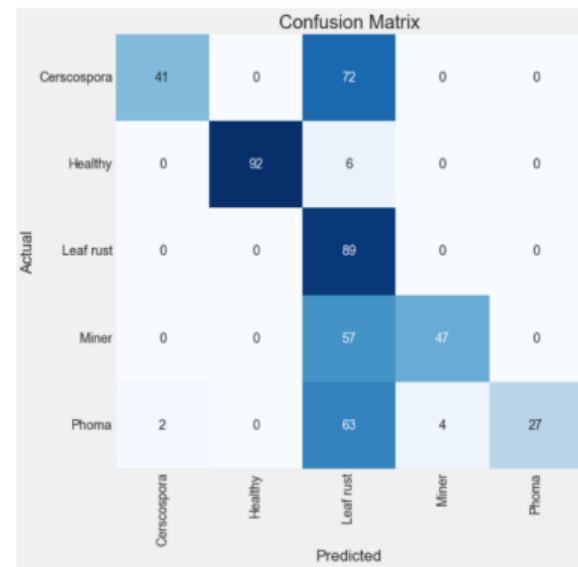


Resultados Preliminares

Modelo: Resnet50 Treinada

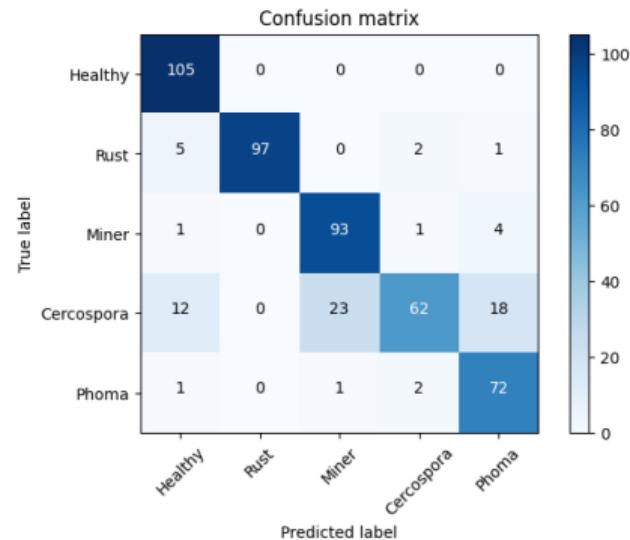


Modelo: Resnet50 de Referência

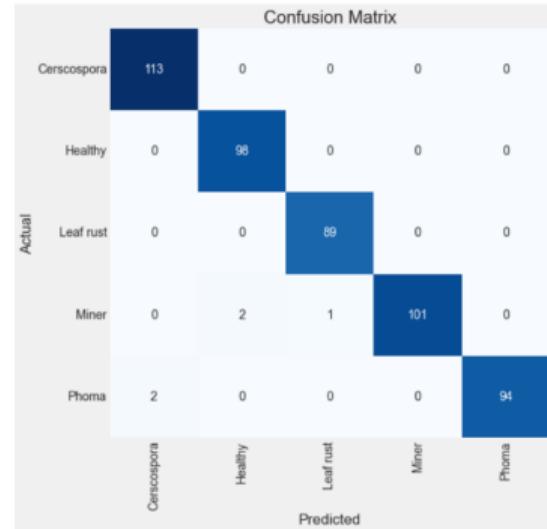


Resultados Preliminares

Modelo: MobileNetV2 Treinada

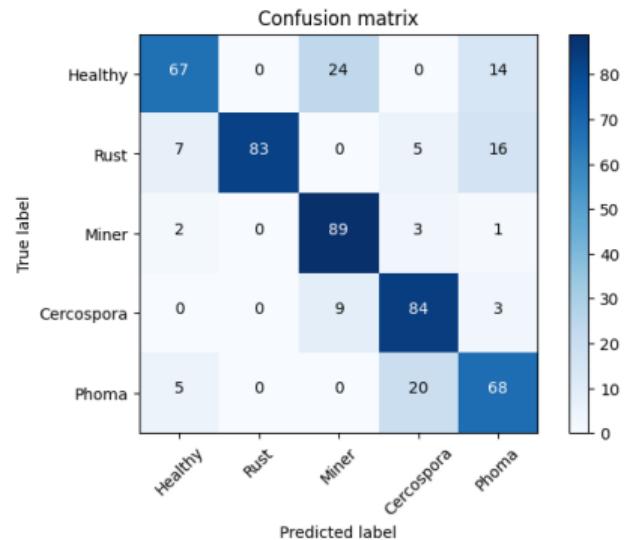


Modelo: MobileNetV2 de Referência

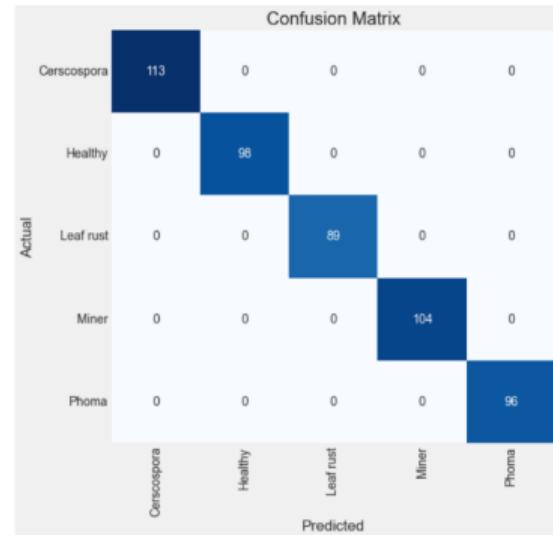


Resultados Preliminares

Modelo: Densenet169 Treinada

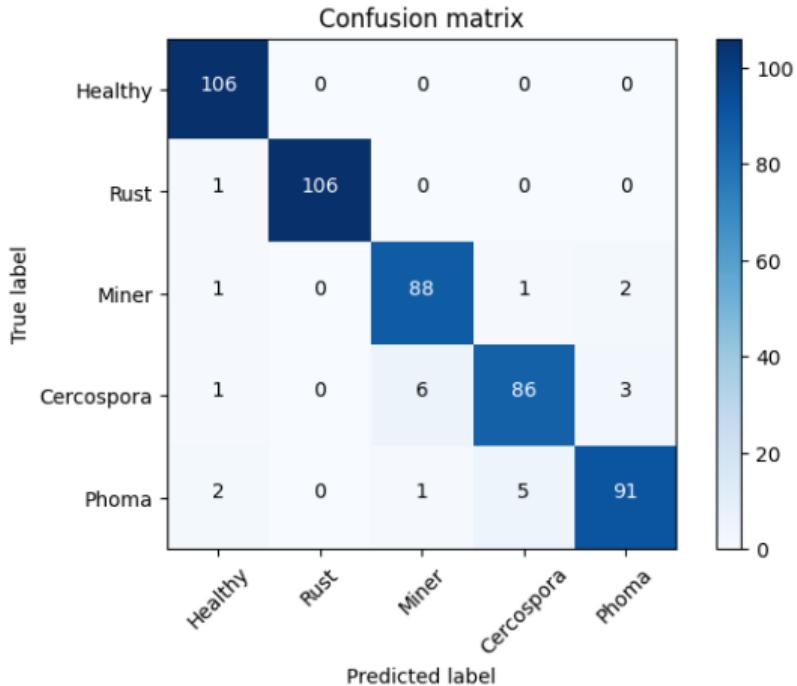


Modelo: Densenet169 de Referência



Resultados Preliminares

Modelo: PavicNet-MC Treinada



Accuracy for Healthy: 1.0 ^a

Accuracy for leaf rust: 0.998 ^b

Accuracy for miner: 0.978

Accuracy for cercospora: 0.968

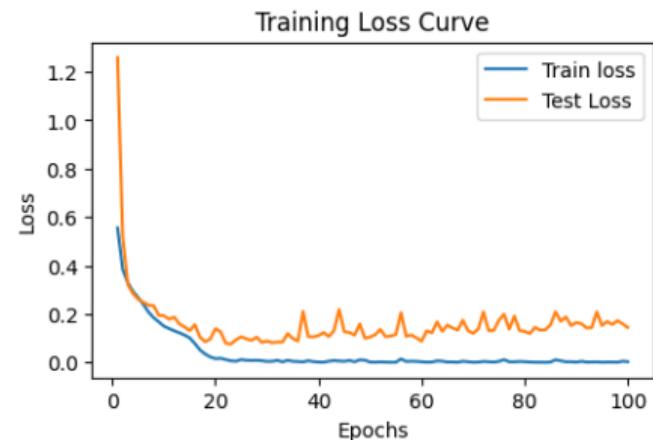
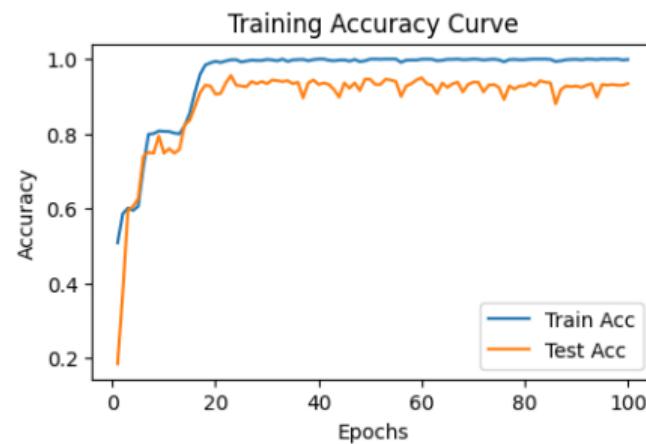
Accuracy for phoma: 0.974

^aClasse com problemas

^bClasse com problemas

Resultados Preliminares

Modelo: Densenet169



Próximos Passos

Curto Prazo:

- Pesquisar e definir novos datasets
- Treinar os modelos implementados com os novos datasets

Médio Prazo:

- Apresentação do artigo no Healthinf
- Dar continuidade a escrita do TCC

Próximos Passos

Curto Prazo:

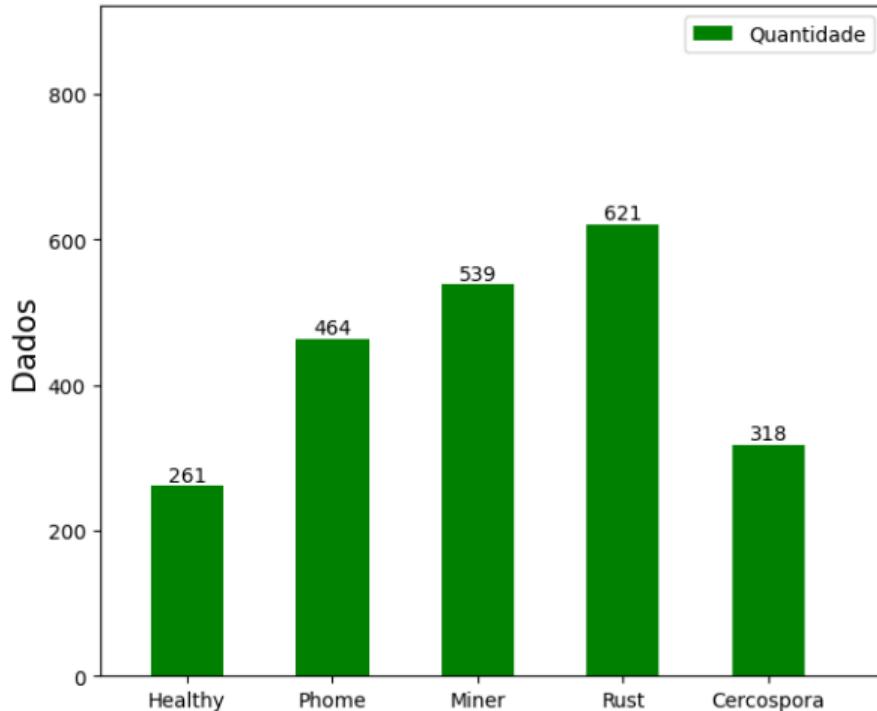
- Pesquisar e definir novos datasets. CONCLUÍDO
- Pré-Processar os novos datasets. CONCLUÍDO
- Treinar os modelos implementados com os novos datasets.

Médio Prazo:

- Apresentação do artigo no Healthinf. CONCLUÍDO
- Dar continuidade a escrita do TCC.

Resultados preliminares

Dataset BRACOL



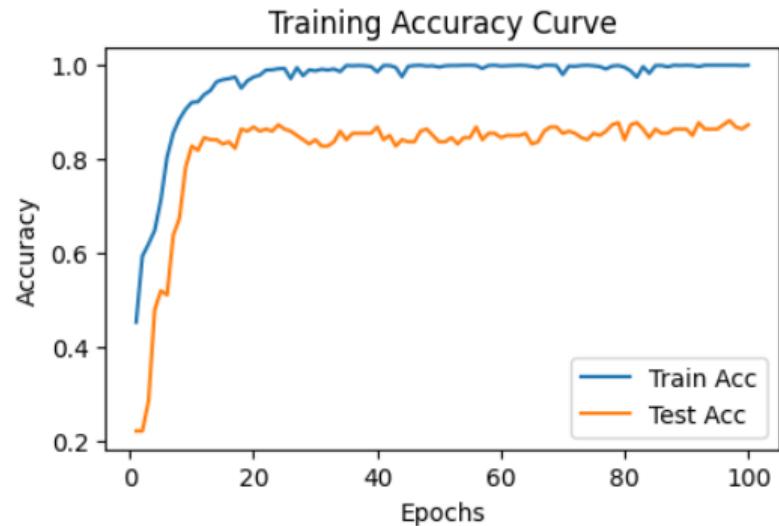
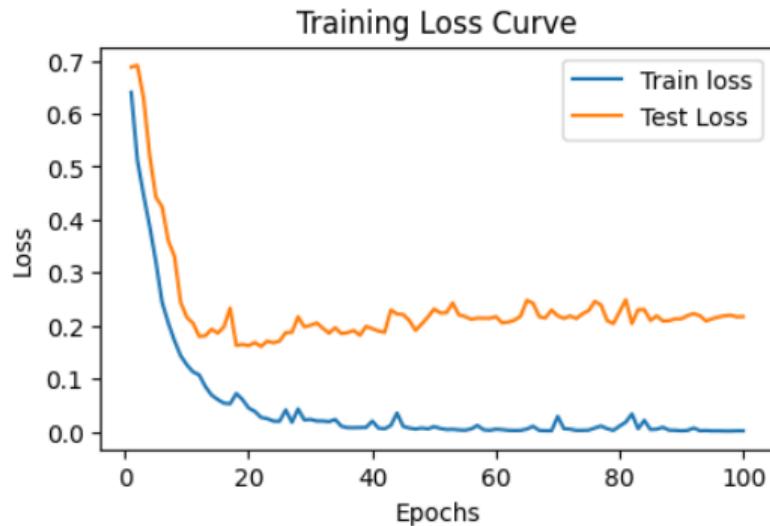
Resultados preliminares

Dataset BRACOL

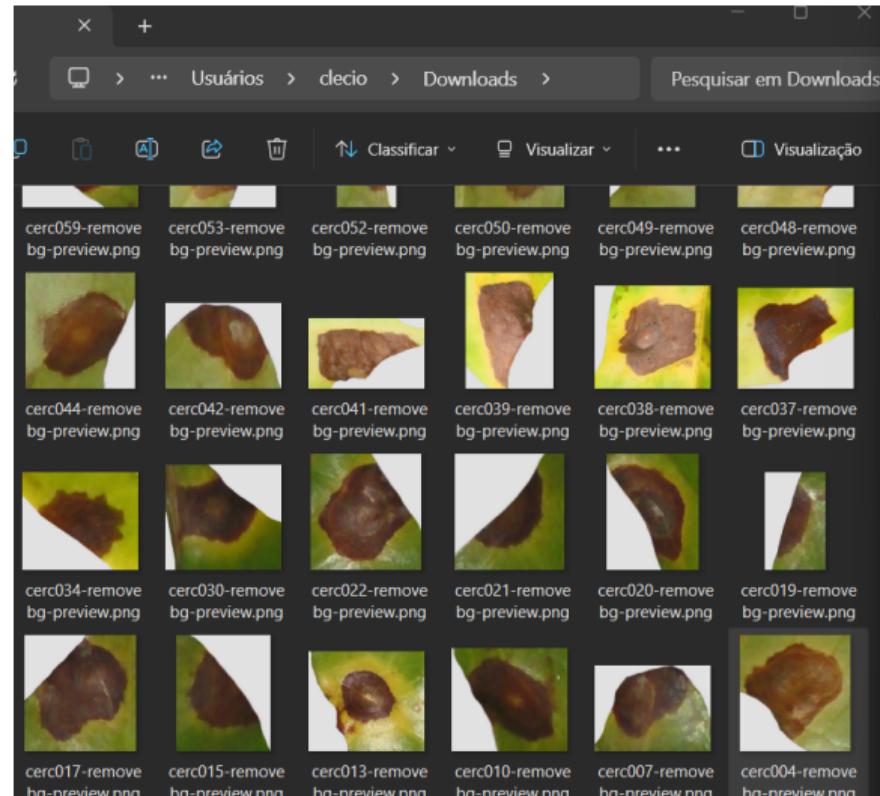


Resultados preliminares

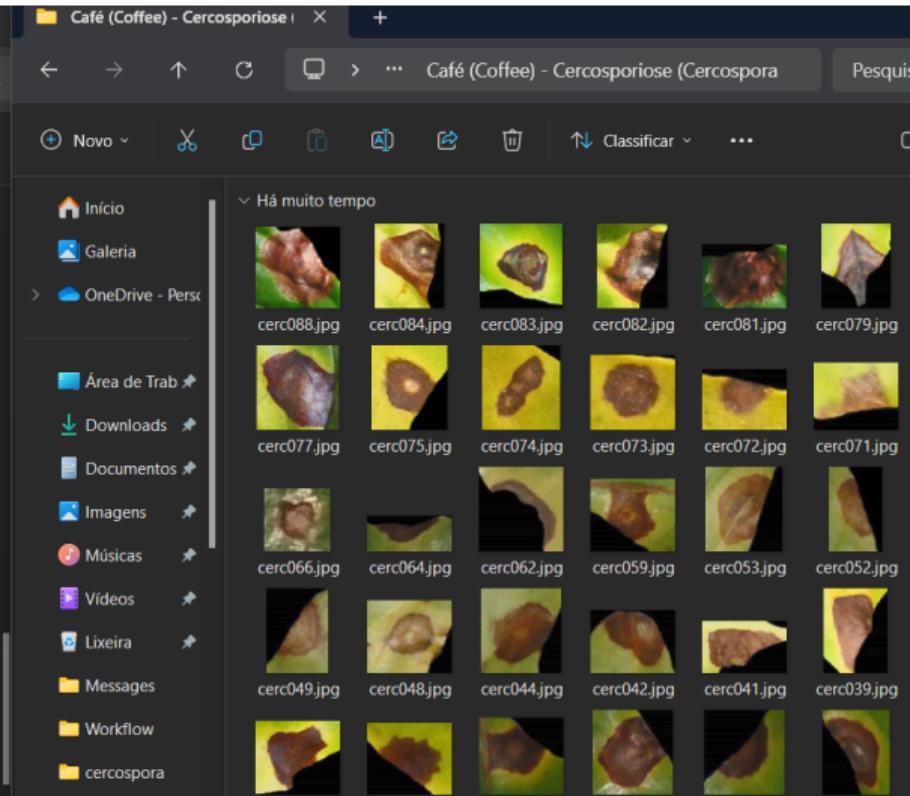
Curvas de treinamento



Resultados preliminares



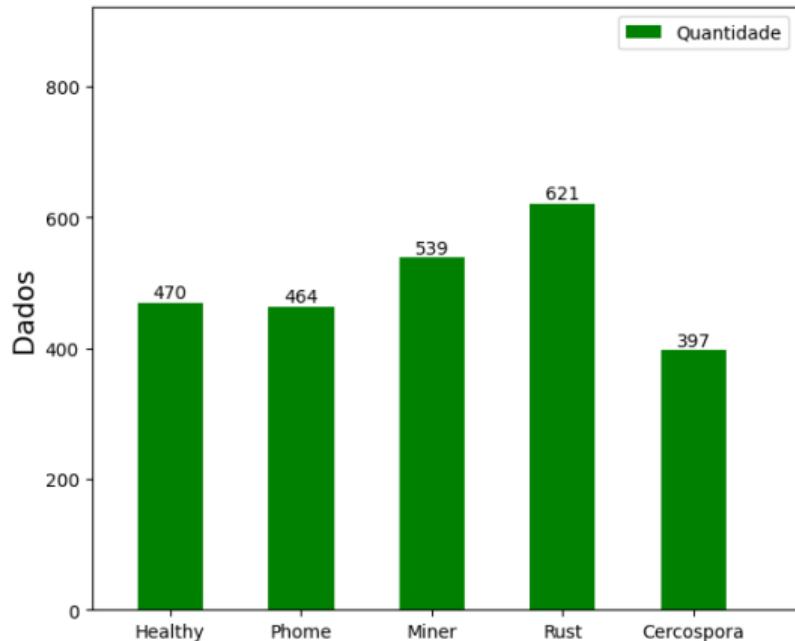
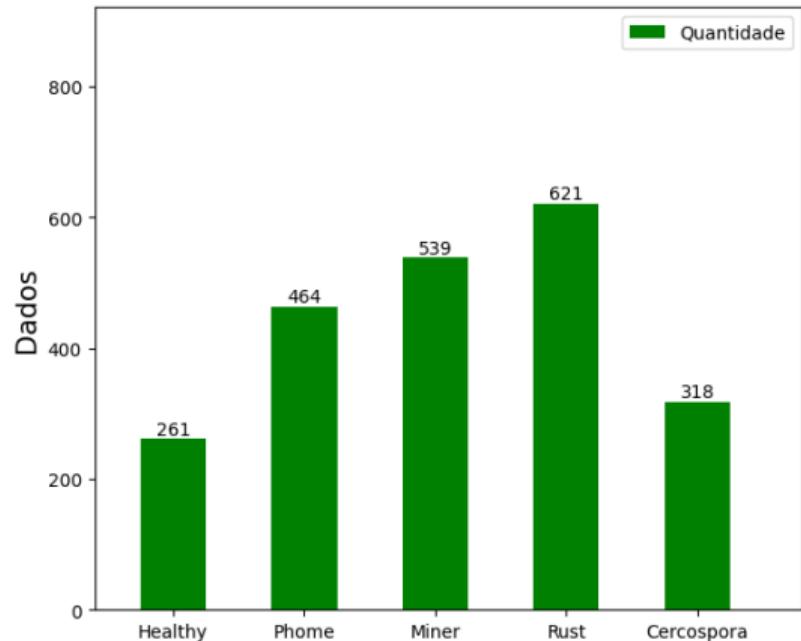
Clécio Elias (UFAC)



Progresso da Pesquisa

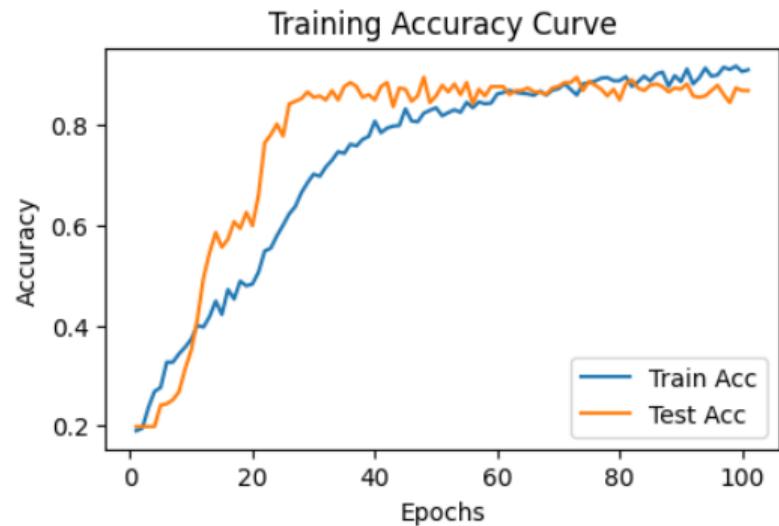
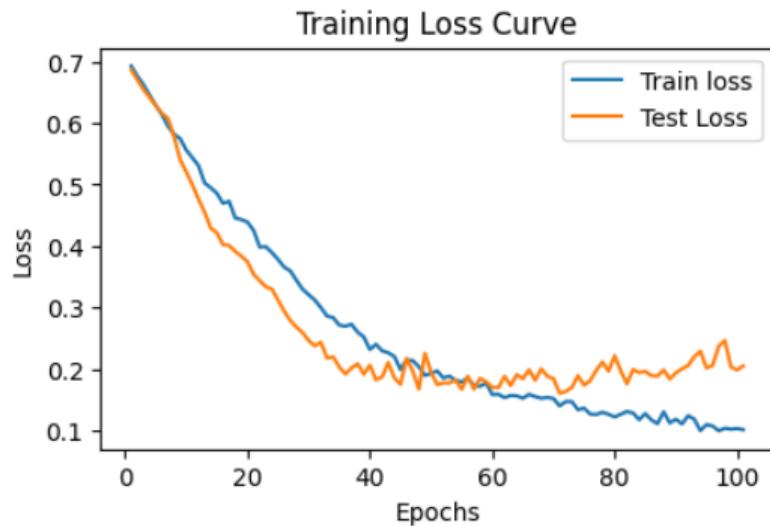
Resultados preliminares

Dataset BRACOL



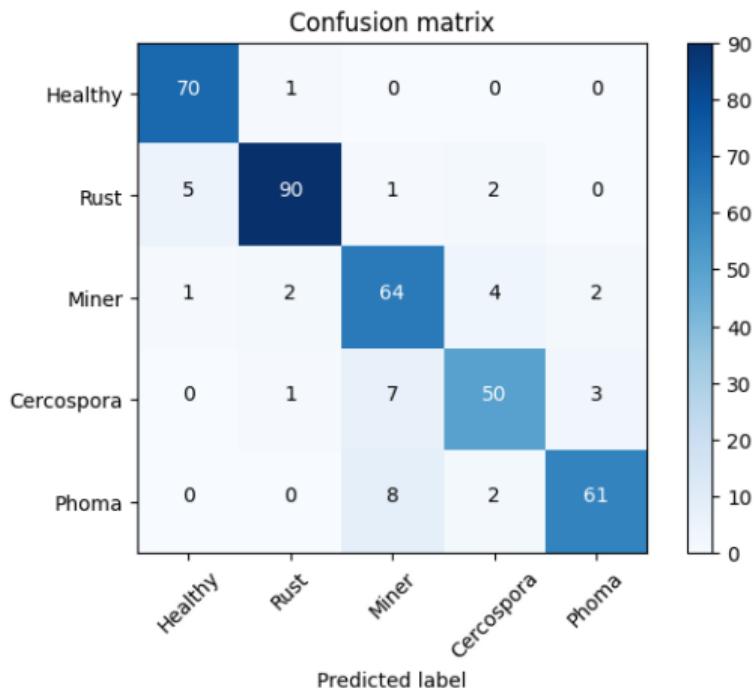
Resultados preliminares

Após ajustes



Resultados preliminares

PavicNet-MC



ACCURACY

The accuracy for Healthy: 0.9813

The accuracy for Rust: 0.9786

The accuracy for Miner: 0.9358

The accuracy for Cercospora: 0.9465

The accuracy for Phoma: 0.9599

Mean Accuracy: 0.9540

PRECISION

The precision for Healthy: 0.9211

The precision for Rust: 0.9592

The precision for Miner: 0.8025

The precision for Cercospora: 0.8254

The precision for Phoma: 0.8784

Mean Precision: 0.8773



Próximos Passos

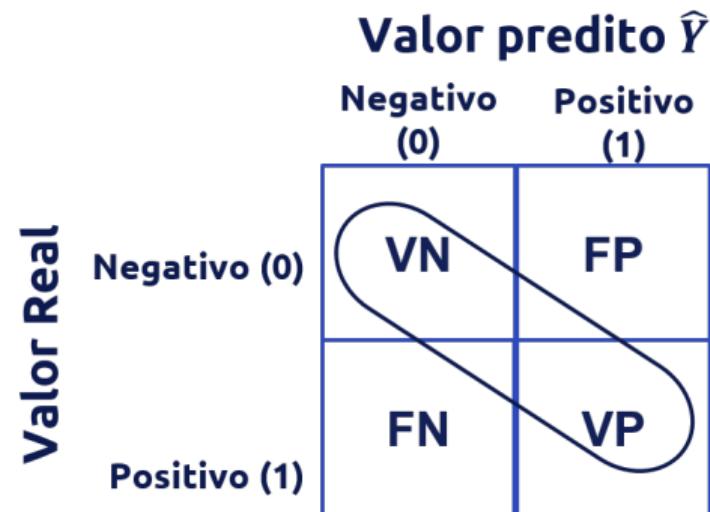
Curto Prazo:

- Pesquisar e definir novos datasets. CONCLUÍDO
- Pré-Processar os novos datasets. CONCLUÍDO
- Continuar o treinamento com o Dataset BRACOL.

Médio Prazo:

- Iniciar o treinamento com os outros datasets.
- Dar continuidade a escrita do TCC.

Resultados preliminares



$$\text{Acurácia} = \frac{VP+VN}{VP+VN+FP+FN}$$

Resultados preliminares

		Classe Predita	
		Positivo	Negativo
Classe Verdadeira	Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Sensibilidade: $VP / (VP + FN)$

Precisão: $VP / (VP + FP)$

Especificidade: $VN / (VN + FP)$

Resultados preliminares

Acurácia

	Accuracy	Dataset	BRACOL	
Classe	PavicNet-MC v5	DenseNet169	ResNet50	InceptionResnetV2
Healthy	0,9813	0,9973	0,9973	0,9973
Rust	0,9599	0,9920	0,9759	0,9893
Miner	0,9278	0,9519	0,9572	0,9733
Cercospora	0,9332	0,9412	0,9439	0,9679
Phoma	0,9599	0,9626	0,9492	0,9813

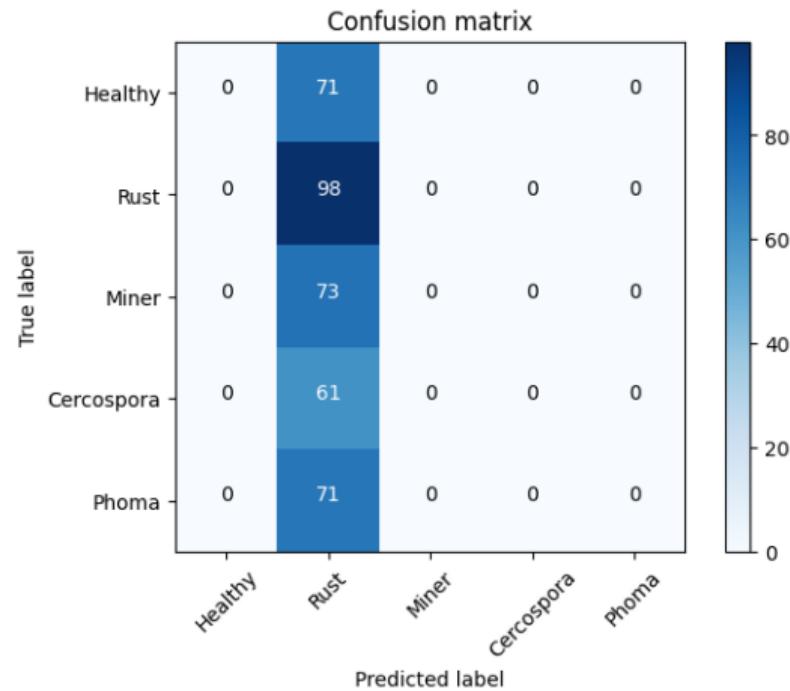
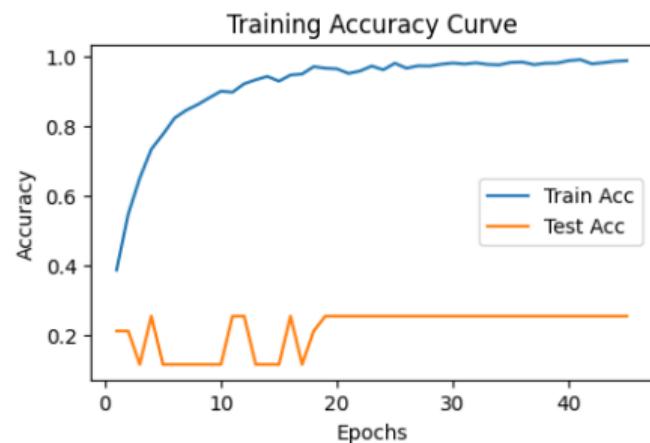
Resultados preliminares

Precisão

	Precision	Dataset	BRACOL	
Classe	PavicNet-MC v5	DenseNet169	ResNet50	InceptionResnetV2
Healthy	0,9211	0,9848	0,9861	1,000
Rust	0,9192	1,000	0,9320	0,9700
Miner	0,8710	0,8710	0,9014	0,9437
Cercospora	0,7308	0,7692	0,8125	0,8769
Phoma	0,8784	0,8889	0,8333	0,9211

Resultados preliminares

MobileNetV2



Resultados preliminares

Training Set

```
N_LABELS = 5
EPOCHS = 300
LR = 0.0001
batch_size = 32 # 32
# Compile the model

# early stop
callback = keras.callbacks.EarlyStopping(
    monitor="val_loss",
    patience=30,
    mode="auto",
    restore_best_weights=True,
    start_from_epoch=0,
)

model_all_0.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=LR),
    #loss='categorical_crossentropy', # categorical cross
    loss='binary_crossentropy', # categorical cross
    metrics=[accuracy'])

history = model_all_0.fit(X_train, y_train,
    batch_size=batch_size,
    epochs=EPOCHS,
    callbacks = [callback],
    validation_data=(X_test, y_test))
```

MobileNetV2 config

MobileNet V2

```
[ ] MobileNetv2 = tf.keras.applications.MobileNetV2(
    input_shape=(224, 224, 3),
    alpha=1.0,
    include_top=True,
    weights=None, #"imagenet",
    input_tensor=None,
    pooling=None,
    classes=5,
    classifier_activation="softmax",
)

model_all_0 = MobileNetv2
```

DenseNet169

```
[ ] DenseNet169 = tf.keras.applications.densenet.DenseNet169(
    input_shape=(224, 224, 3),
    include_top=True,
    weights=None, #"imagenet",
    input_tensor=None,
    pooling=None,
    classes=5,
    classifier_activation="softmax",
)
```

Próximos Passos

Curto Prazo:

- Pesquisar e definir novos datasets. CONCLUÍDO
- Pré-Processar os novos datasets. CONCLUÍDO
- Continuar o treinamento com o Dataset BRACOL. CONCLUÍDO
- Treinar com o dataset JMUBEN em RGB.
- Validar os resultados.

Médio Prazo:

- Iniciar o treinamento com os outros datasets.
- Dar continuidade a escrita do TCC.

Próximos Passos

Curto Prazo:

- Pesquisar e definir novos datasets. CONCLUÍDO
- Pré-Processar os novos datasets. CONCLUÍDO
- Continuar o treinamento com o Dataset BRACOL. CONCLUÍDO
- Treinar com o dataset JMUBEN em RGB. CONCLUÍDO
- Validar os resultados. CONCLUÍDO

Médio Prazo:

- Iniciar o treinamento com os outros datasets.
- Dar continuidade a escrita do TCC.

Resultados preliminares

Acurácia - JMUBEN

	Accuracy	Dataset	JMUBEN		
Classe	PavicNet-MC v1	DenseNet169	ResNet50	InceptionResnetV2	MobileNetv2
Healthy	1.0000	1.0000	1.0000	1.0000	0,7947
Rust	0.9960	1.0000	1.0000	1.0000	0,7800
Miner	0.9827	0.9960	0.9920	0.9987	0,7813
Cercospora	0.9907	1.0000	0.9987	0.9987	0.1680
Phoma	0.9827	0.9960	0.9947	0.9973	0.1880

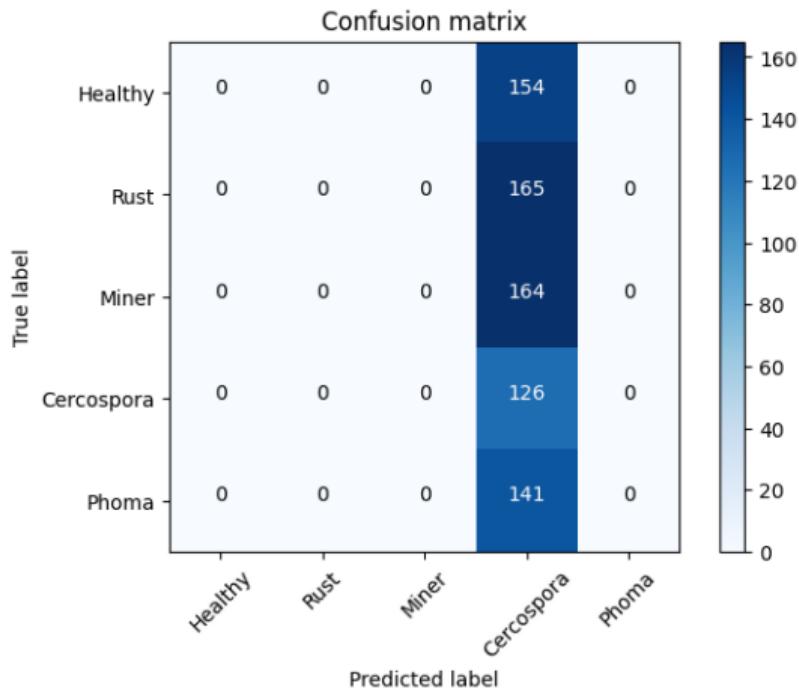
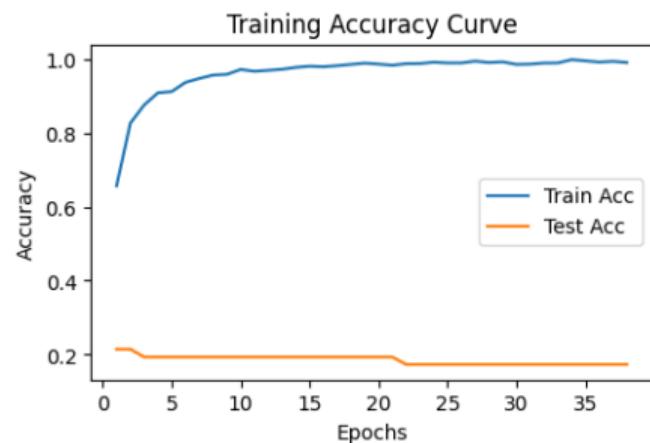
Resultados preliminares

Precisão - JMUBEN

Classe	Precision	Dataset	JMUBEN		
	PavicNet-MC v1		ResNet50	InceptionResnetV2	MobileNetv2
Healthy	1.0000	1.0000	1.0000	1.0000	0
Rust	0.9880	1.0000	1.0000	1.0000	0
Miner	0.9467	0.9939	0.9702	1.0000	0
Cercospora	0.9760	1.0000	1.0000	0.9921	0.1680
Phoma	0.9638	0.9859	0.9928	0.9860	0.1880

Resultados preliminares

MobileNetV2



Próximos Passos

Curto Prazo:

- Treinar com o dataset JMUBEN em RGB. CONCLUÍDO
- Validar os resultados. CONCLUÍDO
- Melhorar os resultados da PavicNet-MC.
- Implementar a arquitetura ShuffleNet.
- Pre-Processar os datasets RoCoLe e Diseases in coffee leafs.

Médio Prazo:

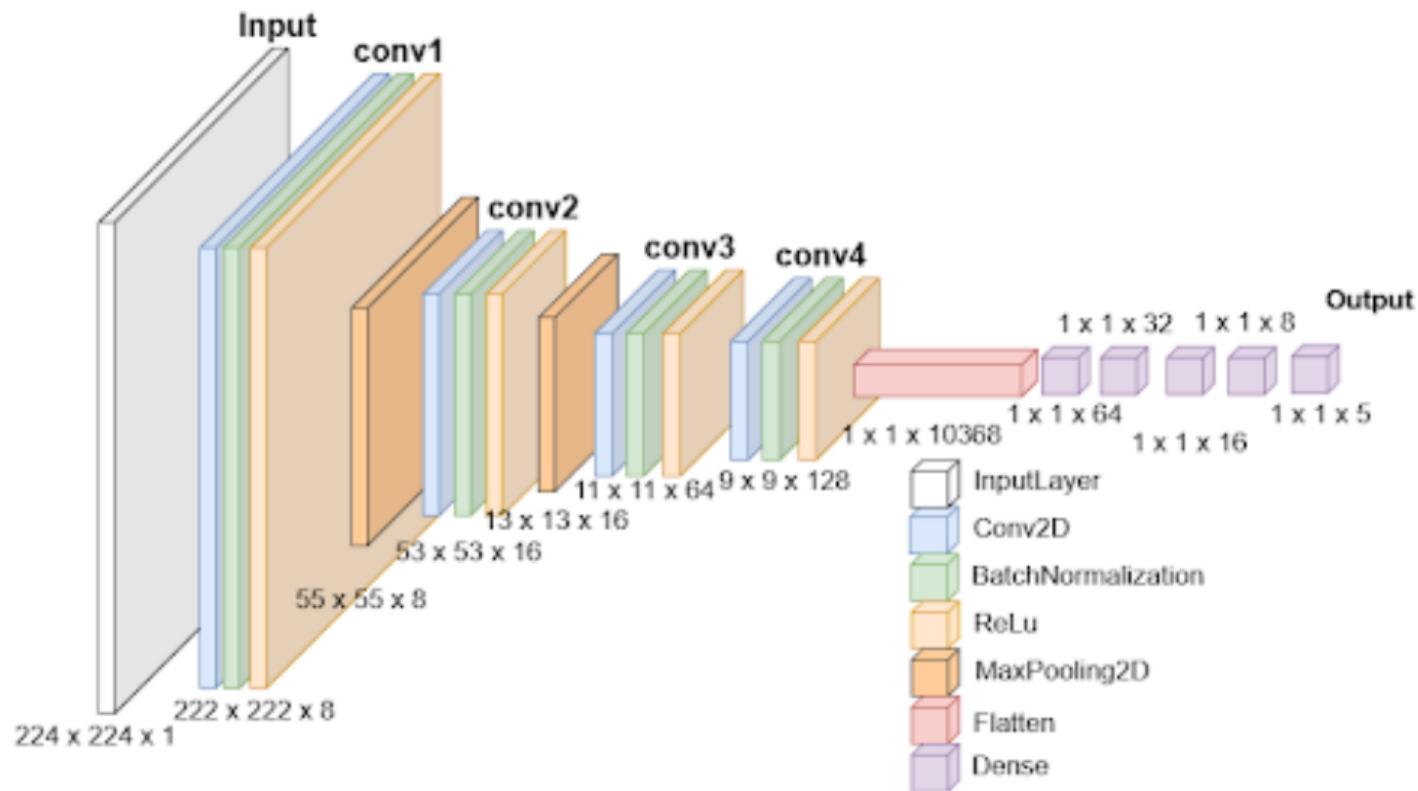
- Iniciar o treinamento com dataset RoCoLe e DisCoLeafs.
- Dar continuidade a escrita do TCC.

Resultados preliminares

Resultados de precisão de classificação de doenças e pragas nas folhas de café utilizando o dataset BRACOL.

Classe	Precision	Dataset	BRACOL	
	PavicNet-MC v5	DenseNet169	ResNet50	InceptionResnetV2
Healthy	0,9211	0,9848	0,9861	1,000
Rust	0,9192	1,000	0,9320	0,9700
Miner	0,8710	0,8710	0,9014	0,9437
Cercospora	0,7308	0,7692	0,8125	0,8769
Phoma	0,8784	0,8889	0,8333	0,9211

Arquitetura PavicNet-MC



Convoluçãoções nas imagens

Input image

sliding window



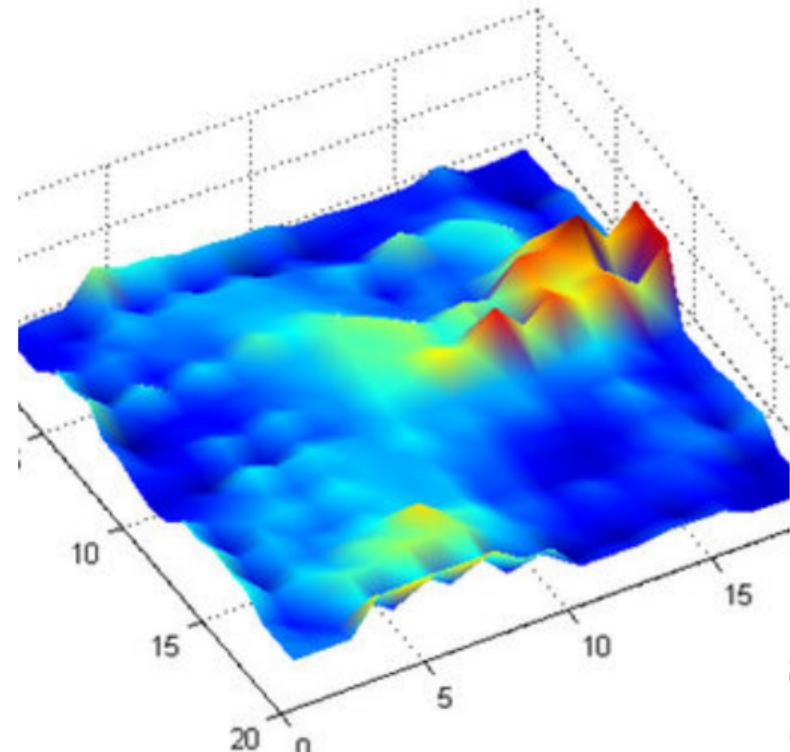
x_{00}	x_{01}	x_{02}	x_{03}	*	*	*	*	*	x_{09}
x_{10}	x_{11}	x_{12}	x_{13}	*	*	*	*	*	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	*	*	*	*	*	x_{29}
x_{30}	x_{31}	x_{32}	x_{33}	*	*	*	*	*	x_{39}
*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*
x_{90}	x_{91}	x_{92}	x_{93}	*	*	*	*	*	x_{99}

Filter

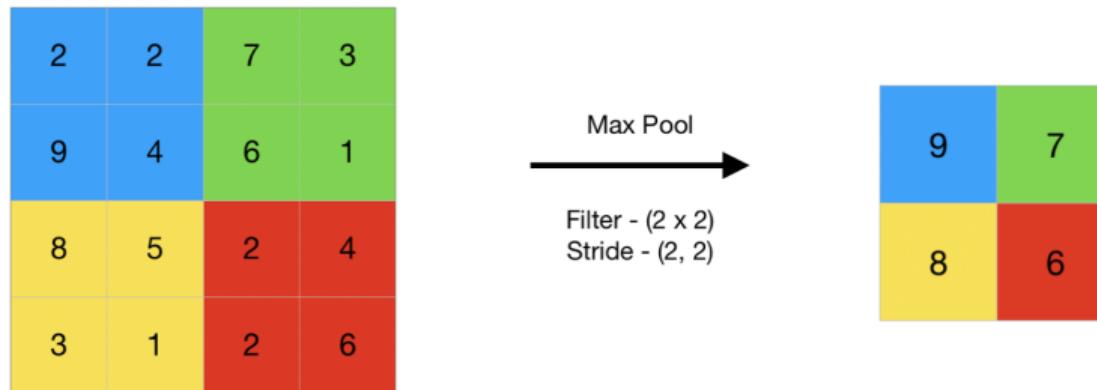
w_{00}	w_{01}	w_{02}
w_{10}	w_{11}	w_{12}
w_{20}	w_{21}	w_{22}

Feature map

y_{00}	y_{01}	y_{02}	y_{03}	*	*	*	y_{07}
y_{10}	y_{11}	y_{12}	y_{13}	*	*	*	y_{17}
y_{20}	y_{21}	y_{22}	y_{23}	*	*	*	y_{27}
y_{30}	y_{31}	y_{32}	y_{33}	*	*	*	y_{37}
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
y_{70}	y_{71}	y_{72}	y_{73}	*	*	*	y_{77}



Camada de MaxPooling (Camada de redução de dimensionalidade dos dados)



Arquitetura PavicNet-MC

Nova PavicNet-MC:

- Modificações nas camadas de convolução e pooling
- Novas camadas de dropout
- De 755.989 para 519.173 parâmetros



Arquitetura ShuffleNet

```
def channel_shuffle(x, groups):
    _, width, height, channels = x.get_shape().as_list()
    group_ch = channels // groups

    x = Reshape([width, height, group_ch, groups])(x)
    x = Permute([1, 2, 4, 3])(x)
    x = Reshape([width, height, channels])(x)
    return x

def shuffle_unit(x, groups, channels,strides):
    y = x
    x = Conv2D(channels//4, kernel_size = 1, strides = (1,1),padding = 'same', groups=groups)(x)
    x = BatchNormalization()(x)
    x = ReLU()(x)

    x = channel_shuffle(x, groups)

    x = DepthwiseConv2D(kernel_size = (3,3), strides = strides, padding = 'same')(x)
    x = BatchNormalization()(x)

    if strides == (2,2):
        channels = channels - y.shape[-1]
        x = Conv2D(channels, kernel_size = 1, strides = (1,1),padding = 'same', groups=groups)(x)
        x = BatchNormalization()(x)

    if strides ==(1,1):
        x =Add()([x,y])
    if strides == (2,2):
        y = AvgPool2D((3,3), strides = (2,2), padding = 'same')(y)
        x = concatenate([x,y])

    x = ReLU()(x)

    return x
```

```
def Shuffle_Net(nclasses, start_channels ,input_shape = (224,224,3)):

    groups = 2
    input = Input (input_shape)

    x = Conv2D (24,kernel_size=3,strides = (2,2), padding = 'same', use_bias = True)(input)
    x = BatchNormalization()(x)
    x = ReLU()(x)

    x = MaxPool2D (pool_size=(3,3), strides = 2, padding='same')(x)

    repetitions = [3,7,3]

    for i,repetition in enumerate(repetitions):
        channels = start_channels * (2**i)

        x = shuffle_unit(x, groups, channels,strides = (2,2))

        for i in range(repetition):
            x = shuffle_unit(x, groups, channels,strides=(1,1))

    x = GlobalAveragePooling2D()(x)

    output = Dense(nclasses,activation='softmax')(x)

    model = Model(input, output)
    return model

[ ] n_classes = 5
input_shape = (224,224,3)
start_channels = 600

model_all_0 = Shuffle_Net(n_classes, start_channels,input_shape)
```

Resultados preliminares

Novos resultados de precisão de classificação de doenças e pragas nas folhas de café utilizando o dataset BRACOL com a nova arquitetura PavicNet-MC e ShuffleNet implementadas.

	Accuracy	Dataset	BRACOL				
Classe	PavicNet-MC v5	DenseNet169	ResNet50	InceptionResnetV2	MobileNetv2	ShuffleNet	PavicNetV7
Healthy	0,9813	0,9973	0,9973	0,9973	0,8102	0,9947	0,9947
Rust	0,9599	0,9920	0,9759	0,9893	0,2620	0,9733	0,9866
Miner	0,9278	0,9519	0,9572	0,9733	0,1952	0,9198	0,9332
Cercospora	0,9332	0,9412	0,9439	0,9679	0,1631	0,9144	0,9492
Phoma	0,9599	0,9626	0,9492	0,9813	0,8102	0,9545	0,9626
MEDIA	0,9524	0,9690	0,9647	0,9818	0,4481	0,9513	0,9653
Parametros	755.989	12.651.205	23.597.957	54.344.421	2.264.389	966.853	519.173

Resultados preliminares

Importância da quantidade reduzida de parâmetros:

- Velocidade no treinamento do modelo
- Velocidade na classificação (utilização do modelo)
- Necessidade de um baixo custo computacional e memória (como dispositivos móveis, drones e etc)



Próximos Passos

Curto Prazo:

- Treinar com o dataset JMUBEN em RGB. CONCLUÍDO
- Validar os resultados. CONCLUÍDO
- Melhorar os resultados da PavicNet-MC. CONCLUÍDO
- Implementar a arquitetura ShuffleNet. CONCLUÍDO
- Pre-Processar os datasets RoCoLe e Diseases in coffee leafs.

Médio Prazo:

- Iniciar o treinamento com dataset RoCoLe e DisCoLeafs.
- Dar continuidade a escrita do TCC.

Resultados preliminares

Diseases in Coffe Leaves (2021):

- 285 imagens da classe rust
- 257 imagens da classe miner
- Arquivo xml separado para cada imagem



Resultados preliminares

RoCoLE (2019):

- 602 imagens da classe rust
- 791 imagens da classe healthy
- 167 imagens da classe red spider mite
- Diversos arquivos de marcação para todas as images (json, xml, xlsx, etc)



Próximos Passos

Curto Prazo:

- Melhorar os resultados da PavicNet-MC. CONCLUÍDO
- Implementar a arquitetura ShuffleNet. CONCLUÍDO
- Pre-Processar o dataset RoCoLe. CANCELADO
- Pre-Processar o dataset Diseases in coffee leafs.

Médio Prazo:

- Iniciar o treinamento com o dataset DisCoLeafs.
- Iniciar treinamentos com o GPUFarm.
- Dar continuidade a escrita do TCC.

Resultados Preliminares

roboflow Workspace Universe Documentation

LEAFCLASSIFICATION
DATASET



[View on Universe](#)

DiseasesInCof... :

Object Detection

Dataset 257

Health Check

Generate

Images [How to Search](#)

Search images

Filter by filename Split Classes Sort By Newest



bicho_mineiro10...



bicho_mineiro118...



bicho_mineiro12...



bicho_mineiro11...



bicho_mineiro10...

Images per page: 50

1 - 50 of 257

Le a Melliha's Computer
VIC LAB

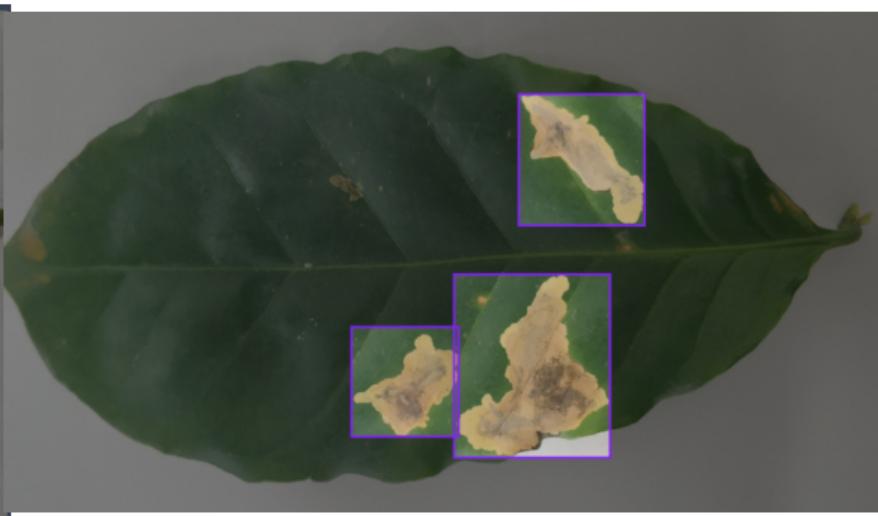
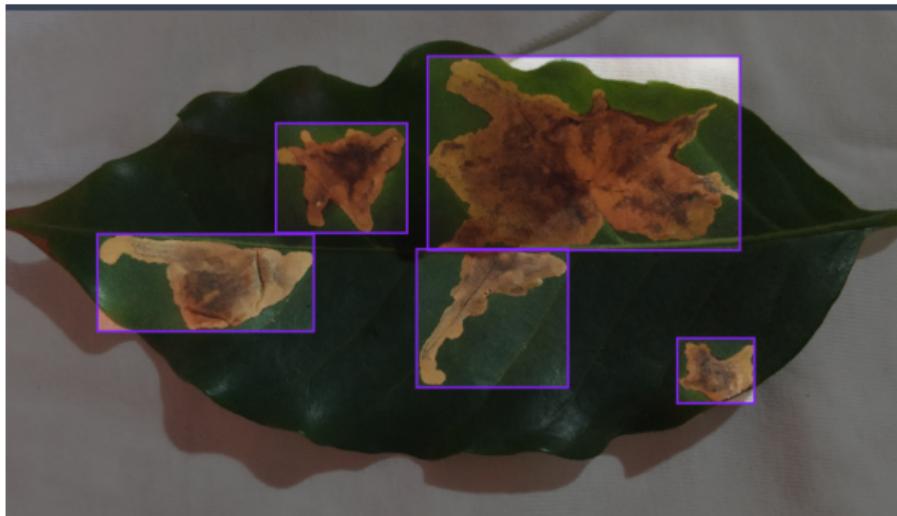
Clécio Elias (UFAC)

Multi - Classification

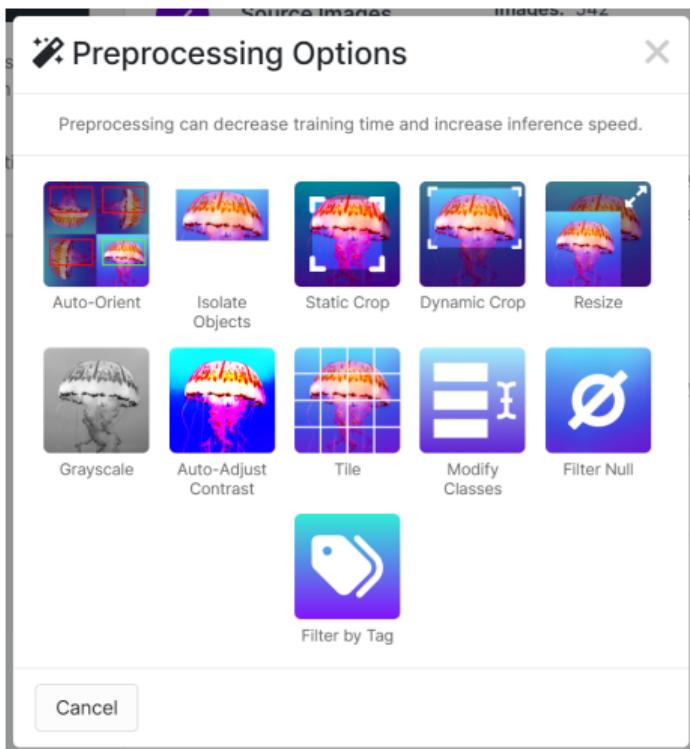
Progresso da Pesquisa

62 / 76

Resultados preliminares



Resultados preliminares



v3 2024-06-20 4:28pm
Generated on Jun 20, 2024

Export Dataset Edit

This version doesn't have a model.
Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

Train with Roboflow Custom Train and Upload
Available Credits: 3

3487 Total Images View All Images →

UI elements at the bottom include navigation arrows, a search icon, and a progress bar labeled 'LAB'.

Próximos Passos

Curto Prazo:

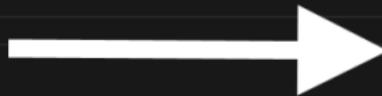
- Melhorar os resultados da PavicNet-MC. CONCLUÍDO
- Implementar a arquitetura ShuffleNet. CONCLUÍDO
- Pre-Processar o dataset RoCoLe. CANCELADO
- Pre-Processar o dataset Diseases in coffee leafs. CONCLUÍDO
- Converter o projeto de treinamento do Tensorflow para Pytorch.

Médio Prazo:

- Iniciar o treinamento com o dataset DisCoLeafs.
- Iniciar treinamentos com o GPUFarm.
- Dar continuidade a escrita do TCC.

Resultados Preliminares

```
for filename in os.listdir(phoma):
    out = '/content/dataset/{}.png'.format(filename.split('.')[0])
    image = cv2.imread(phoma+'/'+filename)
    cv2.imwrite(out, cv2.cvtColor(image, cv2.COLOR_BGR2GRAY))
    img = tf.keras.utils.load_img(out, target_size = (224, 224, 1),color_mode="grayscale")
    img = tf.keras.utils.img_to_array(img)
    img= img/255.
    X_train.append(img)
    y_train.append([0, 0, 0, 1])
X = np.array(X_train)
y = np.array(y_train)
return X, y
```



```
y_train.append(2) # class index for minor
for filename in os.listdir(self.cercospora):
    image = cv2.imread(self.cercospora + '/' + filename)
    image = transform(image)
    X_train.append(image)
    y_train.append(3) # class index for cercospora
for filename in os.listdir(self.phoma):
    image = cv2.imread(self.phoma + '/' + filename)
    image = transform(image)
    X_train.append(image)
    y_train.append(4) # class index for phoma

X = torch.stack(X_train)
y = torch.tensor(y_train)
print('returning X and y data')
return X, y
```

Resultados Preliminares

Resultados - Dataset: Diseases in coffee leaves DCL

Diseases In Coffee Leaf - DCL					
	Accuracy	Dataset	DCL		
Classe	PavicNet-MC v1	DenseNet169	ResNet50	InceptionResnetV2	ShuffleNet
Rust	0.9881	0.9881	1.0000	1.0000	
Miner	0.9881	0.9881	1.0000	1.0000	

Diseases In Coffee Leaf - DCL					
	Precision	Dataset	DCL		
Classe	PavicNet-MC v1	DenseNet169	ResNet50	InceptionResnetV2	ShuffleNet
Rust	1.0000	1.0000	1.0000	1.0000	
Miner	0.9946	0.9946	1.0000	1.0000	

Próximos Passos

Curto Prazo:

- Implementar a arquitetura ShuffleNet. CONCLUÍDO
- Pre-Processar o dataset RoCoLe. CANCELADO
- Pre-Processar o dataset Diseases in coffee leafs. CONCLUÍDO
- Converter o projeto de treinamento do Tensorflow para Pytorch. EM ANDAMENTO
- Treinar modelos com os datasets mesclados.

Médio Prazo:

- Dar inicio a escrita de artigo científico
- Dar continuidade a escrita do TCC.

Próximos Passos

Curto Prazo:

- Finalizar a escrita da seção dos trabalhos relacionados.
- Escrever a seção de metodologia.

Médio Prazo:

- Finzalizar a escrita de artigo científico
- Dar continuidade a escrita do TCC.

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Two-Stage Detection of Diseases and Pests in Coffee Leaves Using Deep Learning

FIRST A. AUTHOR¹, (Fellow, IEEE), SECOND B. AUTHOR², and Third C. Author, Jr.³, (Member, IEEE)

¹National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@boulder.nist.gov)

²Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu)

³Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA

Corresponding author: First A. Author (e-mail: author@ boulder.nist.gov).

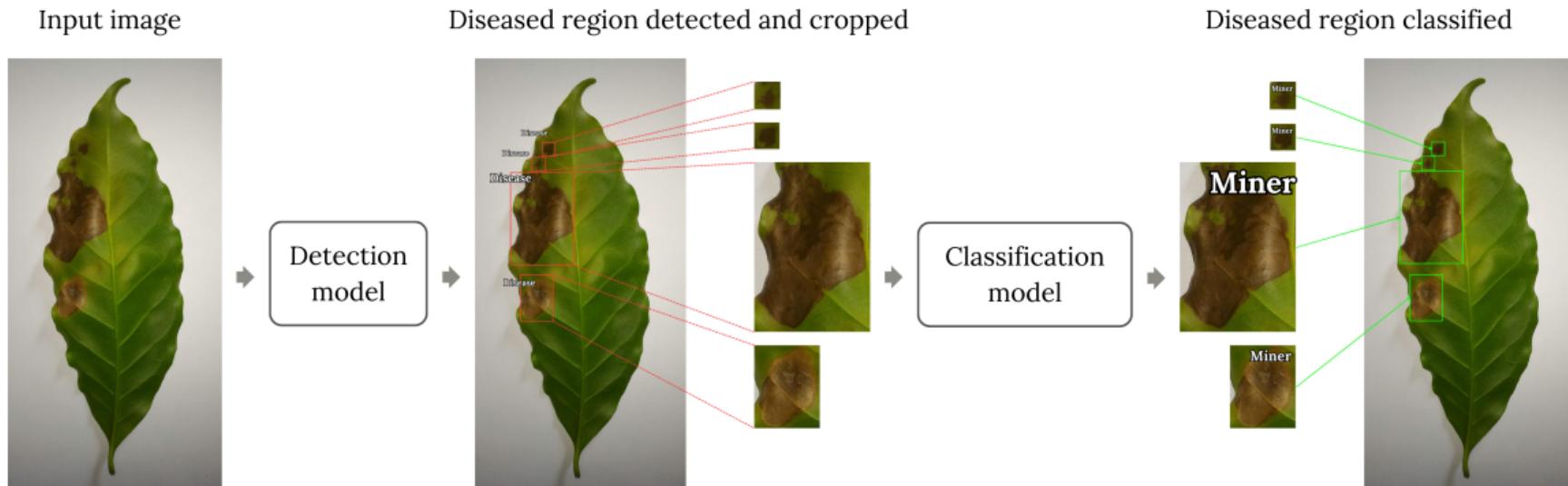
This paragraph of the first footnote will contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

Objetivo

O objetivo deste trabalho é realizar uma detecção em duas etapas das doenças e pragas nas folhas de café. abordando tanto a identificação da região afetada pela doença quanto a classificação da doença específica.



Métodos



Métodos

Modelos de detecção:

- YOLOv8
- YOLOv10

Modelos de classificação:

- DenseNet169
- InceptionResnetV2
- ShuffleNet
- ResNet50

Datasets utilizados

- BRACOL (1747 imagens)
- Diseases and Pests in Coffee Leaves (DPCL) (542 imagens)

Classes:

- Rust
- Miner
- Phoma
- Cercospora
- Healthy*

Exemplos



Métodos

Exemplos



Saudável

Cercospora

Phoma

Rust

Miner



Clécio Elias (UFAC)

Multi - Classification

Progresso da Pesquisa

Próximos Passos

Curto Prazo:

- Finalizar a sessão de Métodologia.
- Iniciar a sessão de resultados.

Médio Prazo:

- Finalizar a escrita de artigo científico
- Dar continuidade a escrita do TCC.