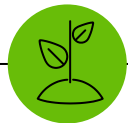


Classificação de estresses bióticos em folhas de café utilizando visão computacional e aprendizado profundo



INSTITUTO
FEDERAL
São Paulo

Câmpus
São João da Boa Vista



Discente: Luís Henrique Vieira – BCC

Orientador: Gabriel Marcelino Alves



1

Contexto e Objetivos



Contexto

- Café no Brasil:
 - Maior produtor e exportador do mundo;
 - 2º maior consumidor do mundo;
 - Estados com maior produção: Minas Gerais, Espírito Santo e São Paulo;

Figura 1 –Pé de café



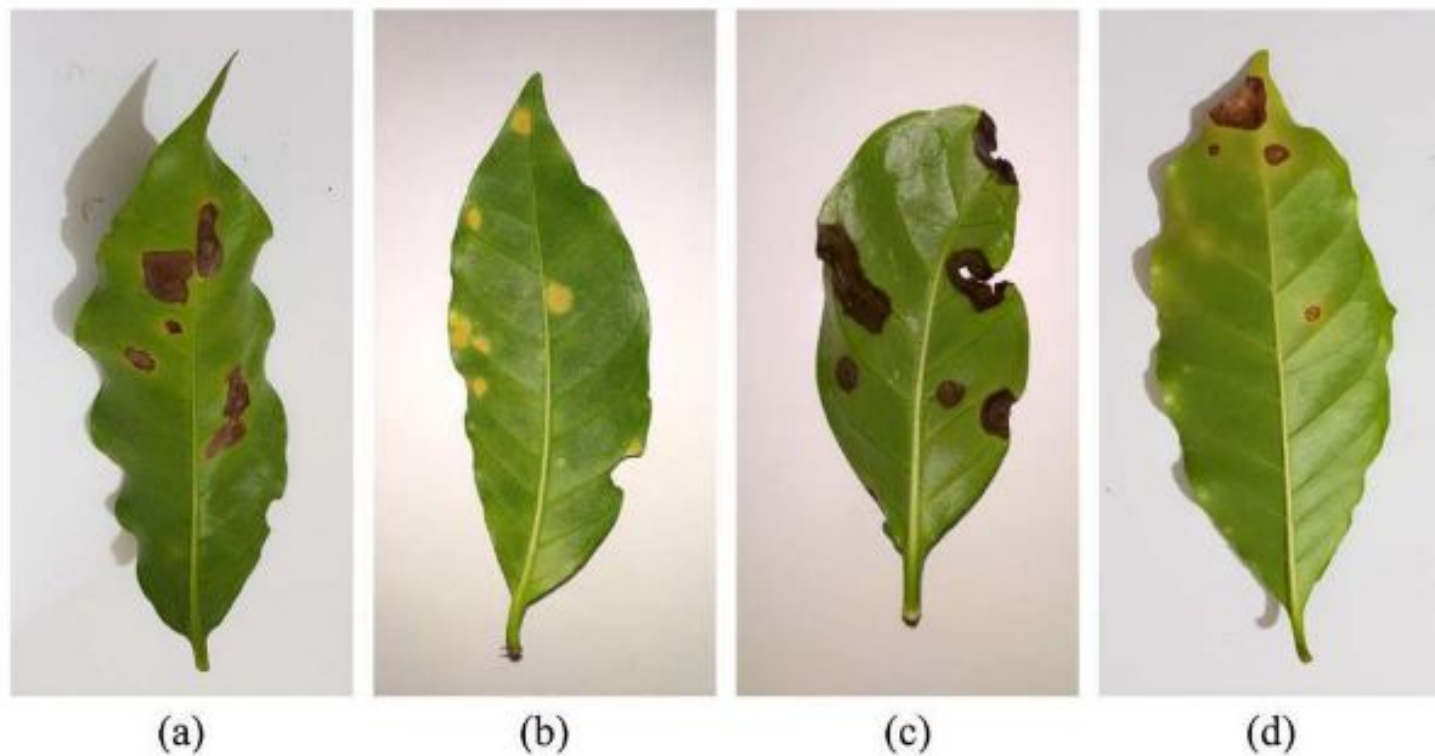
Fonte: Elaborado pelo autor.



Estresses bióticos

- É o dano causado nas plantas por outros seres vivos;
- Os principais são:
 - Ferrugem;
 - Bicho mineiro;
 - Mancha de phoma;
 - Cercosporiose.
- Identificação a partir de coleta, separação e observação

Figura 2 – (a) bicho mineiro, (b) ferrugem, (c) mancha de phoma, (d) cercosporiose.



Fonte: Elaborado por Esgario, Krohling e Ventura (2020)



Objetivos

- Objetivo Geral:
 - Identificar e classificar os estresses bióticos em folhas de café utilizando técnicas de visão computacional e aprendizagem profunda.
- Objetivos Específicos:
 - Implementar redes neurais (CNN);
 - Aplicar e avaliar as redes;
 - Construir base de imagens adicional do leste paulista.

2

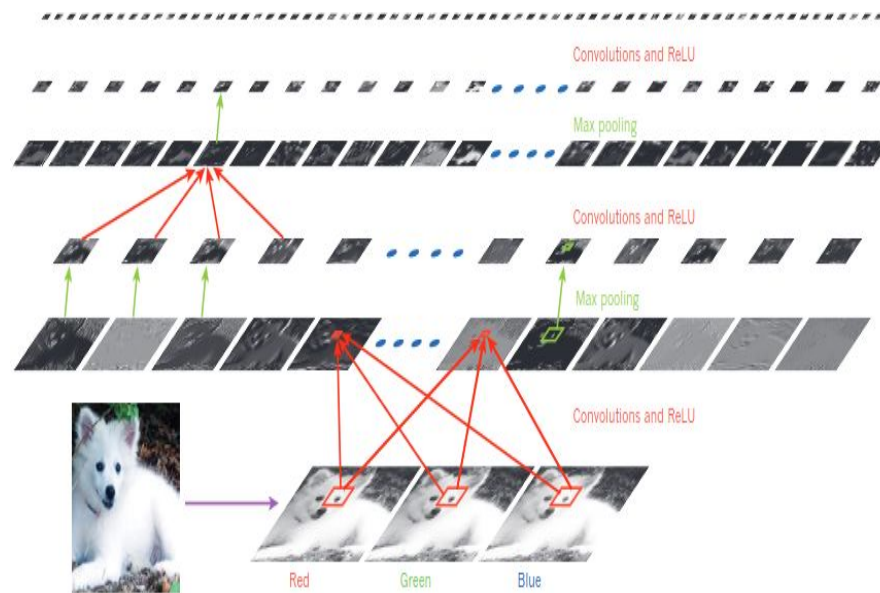
Revisão da Literatura



Aprendizado profundo

- O aprendizado profundo é uma sub-área do aprendizado de máquina;
- Permite a extração de características de dados brutos.

Figura 3 – Exemplo da extração de características.



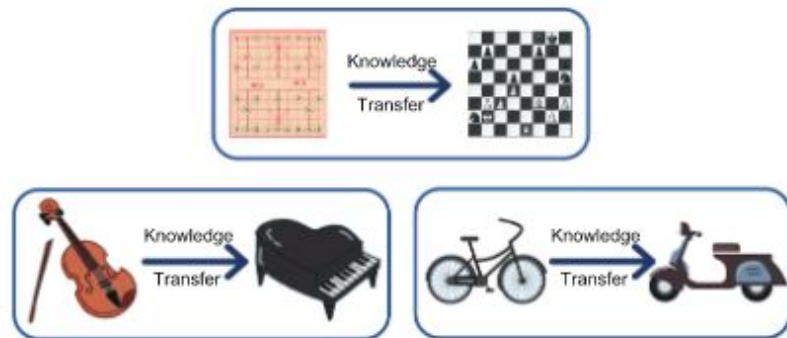
Fonte: Elaborado por LeCun; Bengio; Hinton (2015)



Transferência de aprendizagem

- Transferência entre o domínio de origem e destino;
- Relaxa a necessidade de dados rotulados.

Figura 4 – Exemplos intuitivos de transferência de aprendizagem.



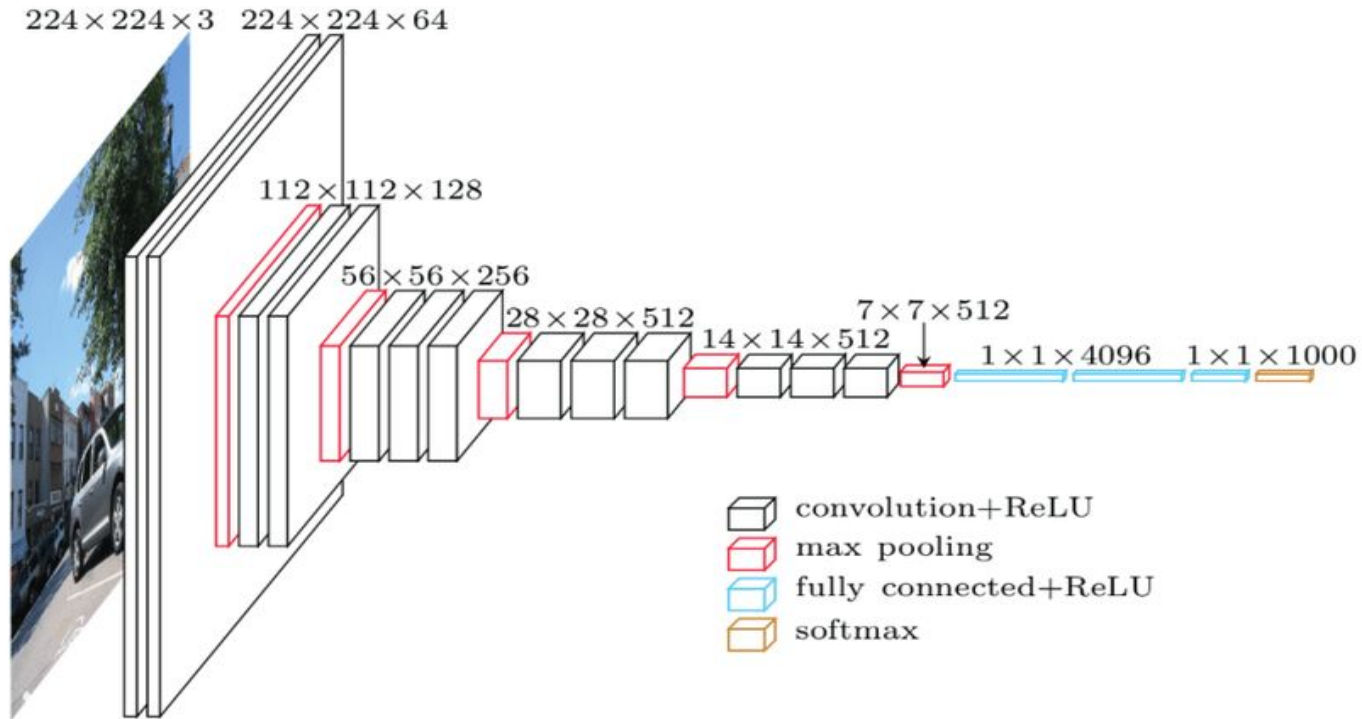
Fonte: (Zhuang et al., 2021)



Rede neural convolucional

- Utilizada no reconhecimento de imagem e voz;
- Arquitetura composta pelas camadas:
 - Convolução;
 - Não linearidade;
 - Pooling;
 - Totalmente conectadas.

Figura 5 – Exemplo da Arquitetura VGG16



Fonte: Elaborado por Bezdan e Džakula (2019).



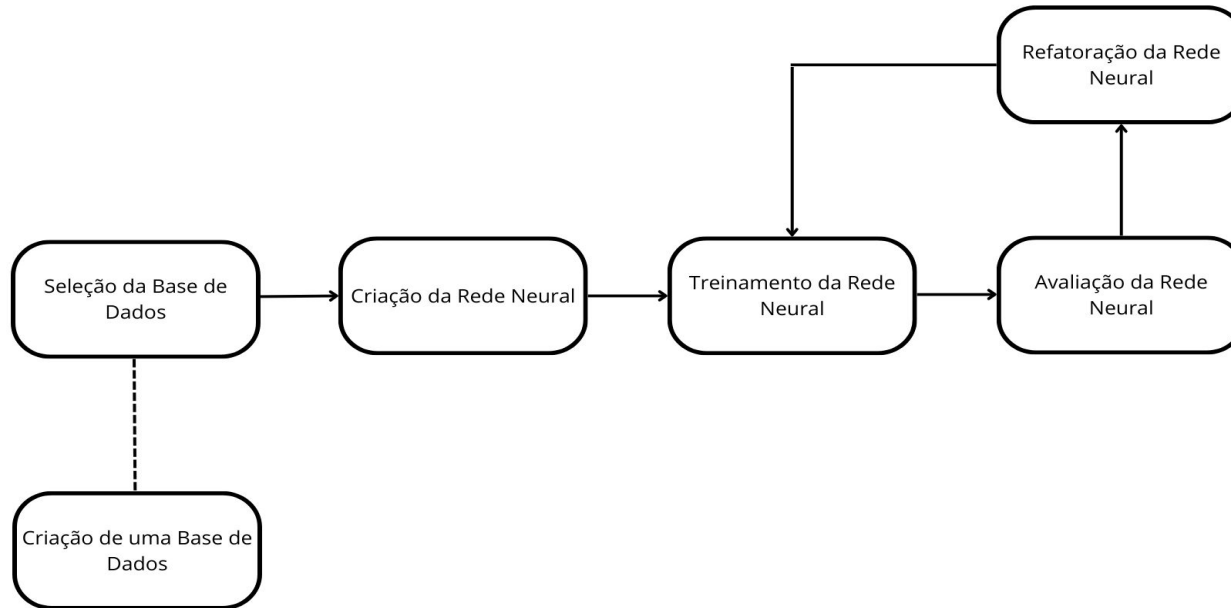
Trabalhos correlatos

- Trabalho de Liu, Zhang, He e Li (2018):
 - Identificação de doenças nas folhas de macieiras, com base na arquitetura AlexNet;
- Trabalho de Esgario, Krohling e Ventura (2020):
 - Detecção de estresses bióticos nas folhas de café;

3

Metodologia

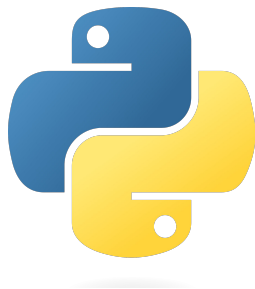
Figura 6 - Ilustração metodologia



Fonte: Elaborado por Bezdan e Džakula (2019).



Ferramentas e Tecnologias





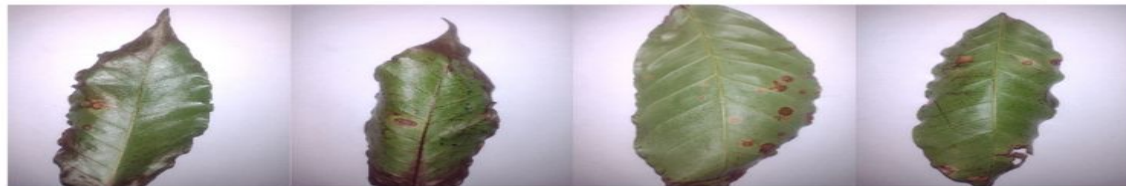
Criação da base de imagens

- Estresses bióticos: Ferrugem e Cercosporiose;
- Total de Folhas: 264;
- Ferrugem (193), Cercosporiose (20) e Saudáveis (51);
- Coletadas:
 - Divinolândia;
 - São Sebastião da Gramma;
 - Caconde;
 - Santo Antônio do Jardim.
- Não utilizada nas redes neurais

Figura 7 - Imagens da base do leste paulista



(a) Saudáveis



(b) Cercosporiose



(c) Ferrugem

Fonte: Elaborado pelo autor.



Conjunto de dados e Construção da rede

- Utilização do conjunto *Symptom* da base de imagens **BRACOL**;
- Processo de *data augmentation*;
- Divisão do conjunto de treinamento (70%), validação (15%) e teste (15%);
- Arquiteturas desenvolvidas:
 - VGG16
 - ResNet50V2
 - MobileNetV2

Figura 8 – Código do método de criação das redes

```
def create_model(self) -> None:

    if self.architecture == Architecture.RESNET50V2:
        from keras.applications.resnet_v2 import ResNet50V2
        model_choose = ResNet50V2(include_top=False,
                                   weights='imagenet', input_shape=self.input_shape)

    elif self.architecture == Architecture.MOBILENETV2:
        from keras.applications.mobilenet_v2 import MobileNetV2
        model_choose = MobileNetV2(include_top=False,
                                    weights='imagenet', input_shape=self.input_shape)

    else:
        from keras.applications.vgg16 import VGG16
        model_choose = VGG16(include_top=False,
                              weights='imagenet', input_shape=self.input_shape)

    if self.fine_tune > 0:
        for layer in model_choose.layers[:self.fine_tune]:
            layer.trainable = False
    else:
        for layer in model_choose.layers:
            layer.trainable = True

    top_model = model_choose.output
    top_model = GlobalAveragePooling2D()(top_model)
    top_model = Dense(4096, activation='relu')(top_model)
    top_model = Dense(1072, activation='relu')(top_model)
    output_layer = Dense(self.n_classes, activation='softmax')(top_model)

    self.model = Model(inputs=model_choose.input, outputs=output_layer)

    self.model.compile(optimizer=self.optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
```

Fonte: Elaborado pelo autor.

Figura 9 – Código de uso método de criação das redes

```
1 model_mobilenet = mobilenet.create_model()
2 model_mobilenet_fine_tune = mobilenet_fine_tune.create_model()
3
4 model_resnet = resnet.create_model()
5 model_resnet_fine_tune = resnet_fine_tune.create_model()
6
7 model_vgg = vgg.create_model()
8 model_vgg_fine_tune = vgg_fine_tune.create_model()
```

Fonte: Elaborado pelo autor.



Hiper parâmetros para treino

Tabela 1 - Hiper parâmetros utilizados

Hiper Parâmetro	Valor
Otimizador	Adam
Função de perda	Entropia cruzada
Tamanho do batch	32
Épocas	100
Taxa de aprendizagem	0,0001

Fonte: Elaborada pelo autor.



Resultados



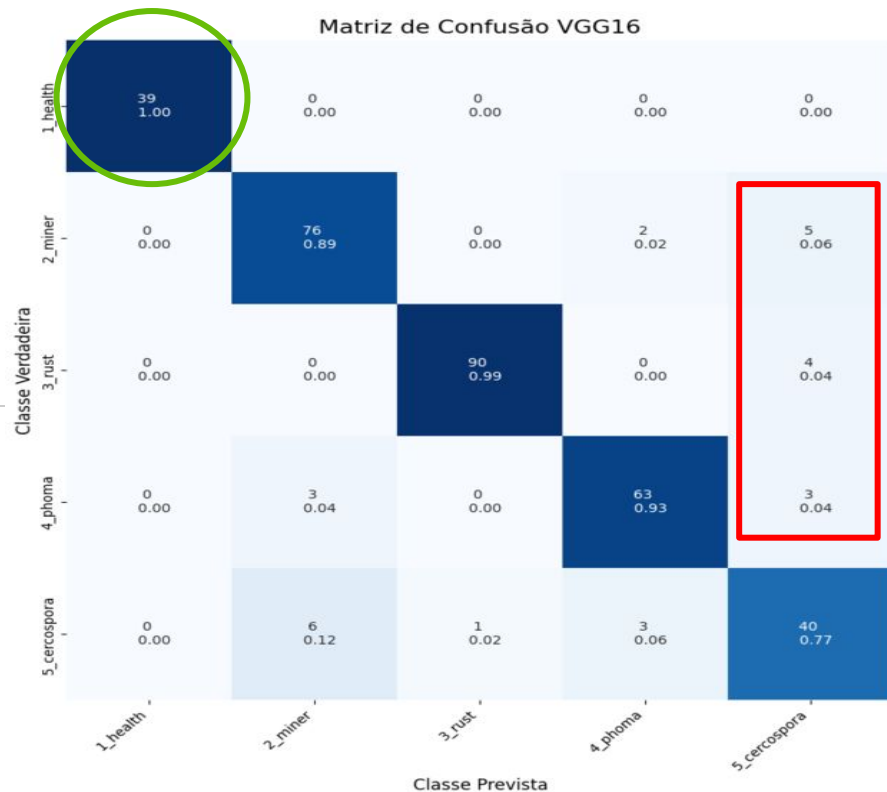
Avaliação dos modelos

Tabela 2 – Resultados das métricas de avaliação.

Arquitetura	Acurácia	Precisão	Recall
ResNet50V2	93,4%	93,4%	93,4%
VGG16	91,9%	92,1%	91,9%
MobileNetV2	89,9%	90,8%	89,9%

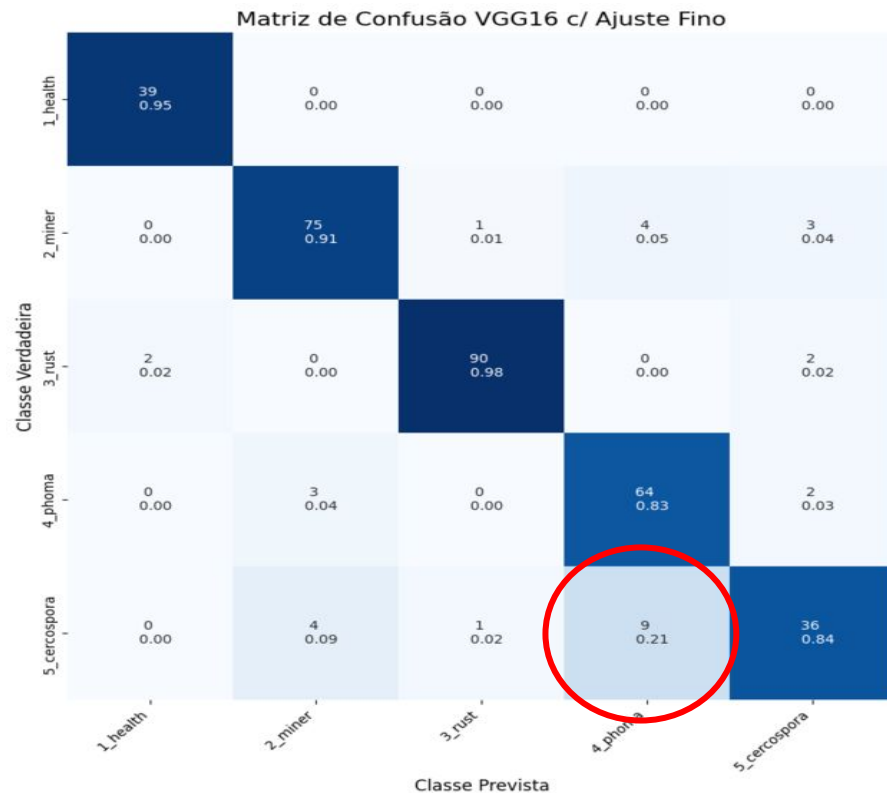
Fonte: Elaborado pelo autor.

Figura 10 - VGG16



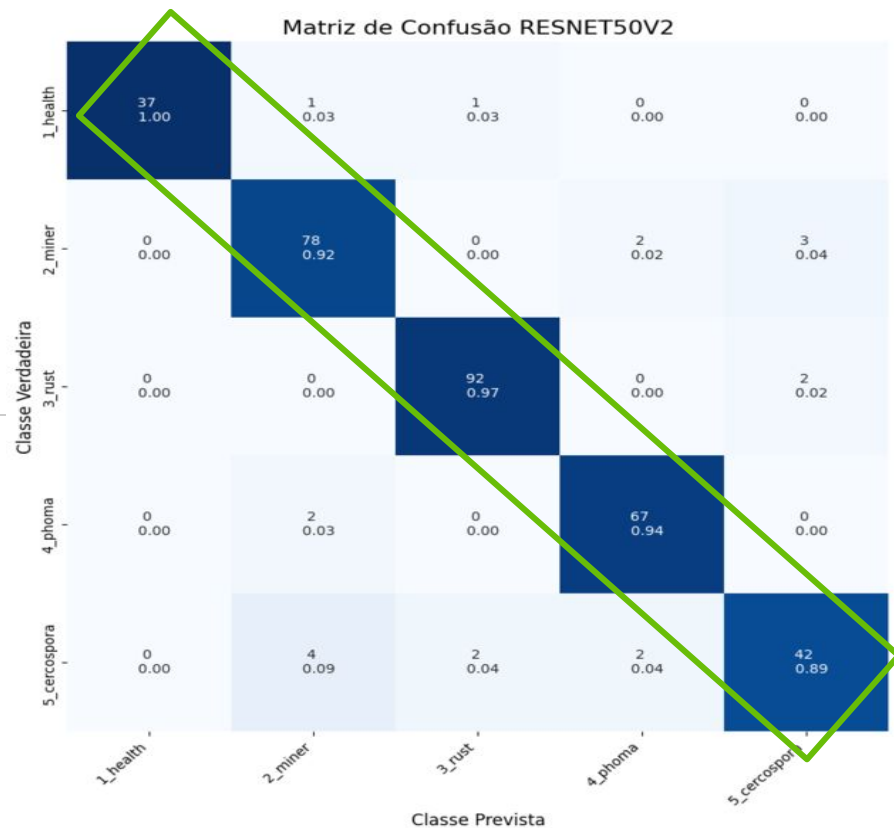
Fonte: Elaborado pelo autor.

Figura 11 - VGG16 com ajuste fino



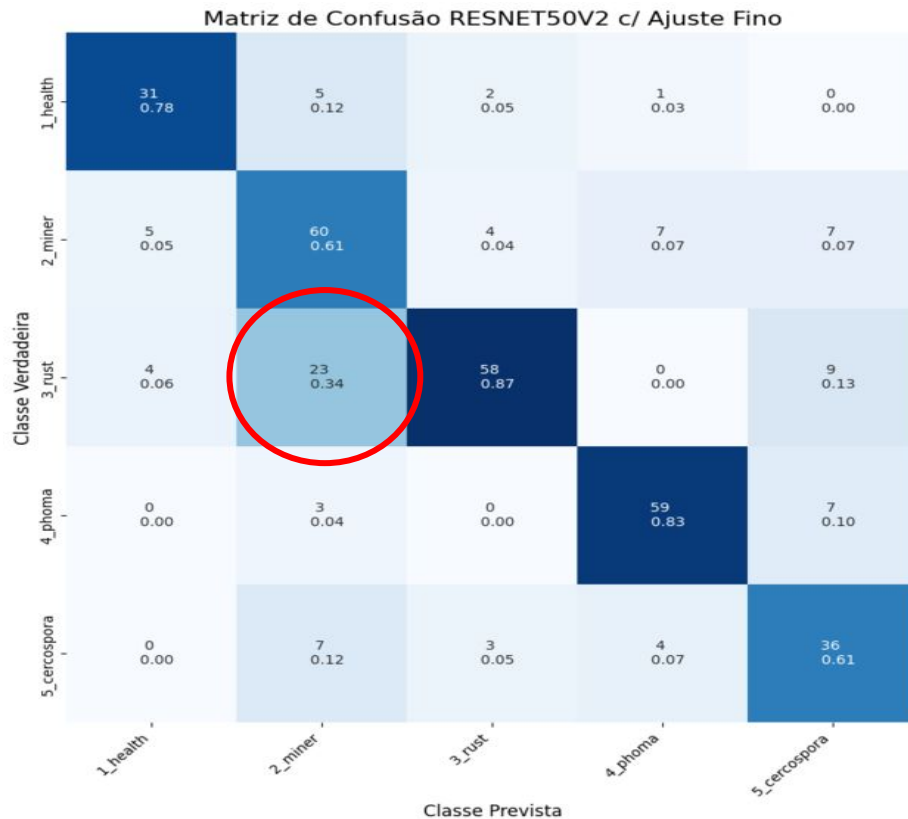
Fonte: Elaborado pelo autor.

Figura 12 - ResNet50V2



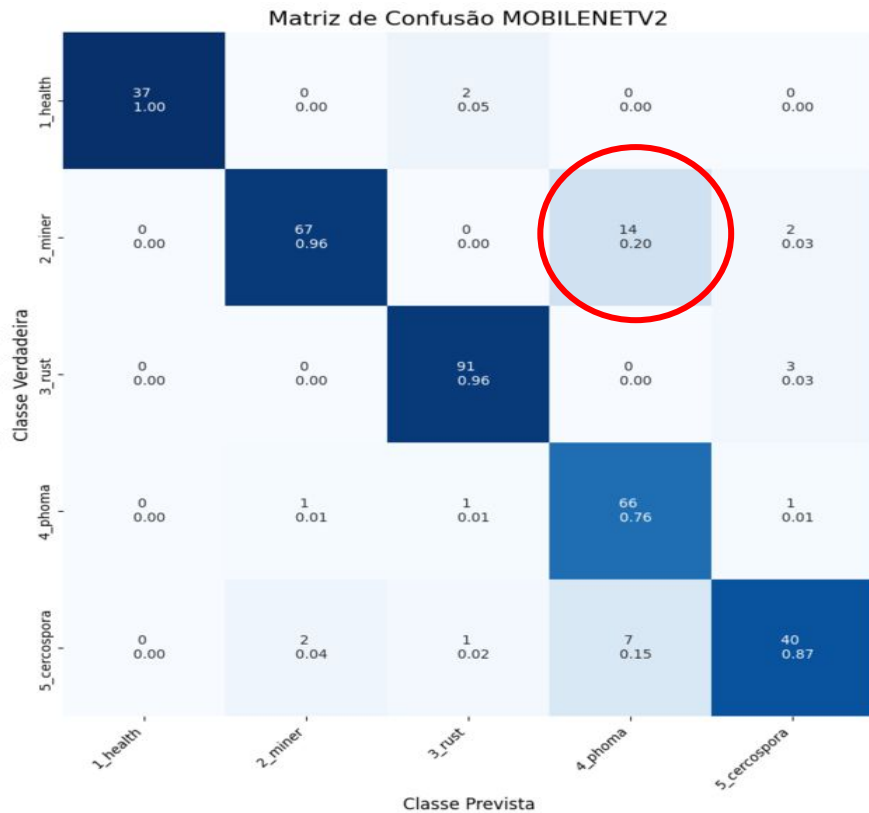
Fonte: Elaborado pelo autor.

Figura 13 - ResNet50V2 com ajuste fino



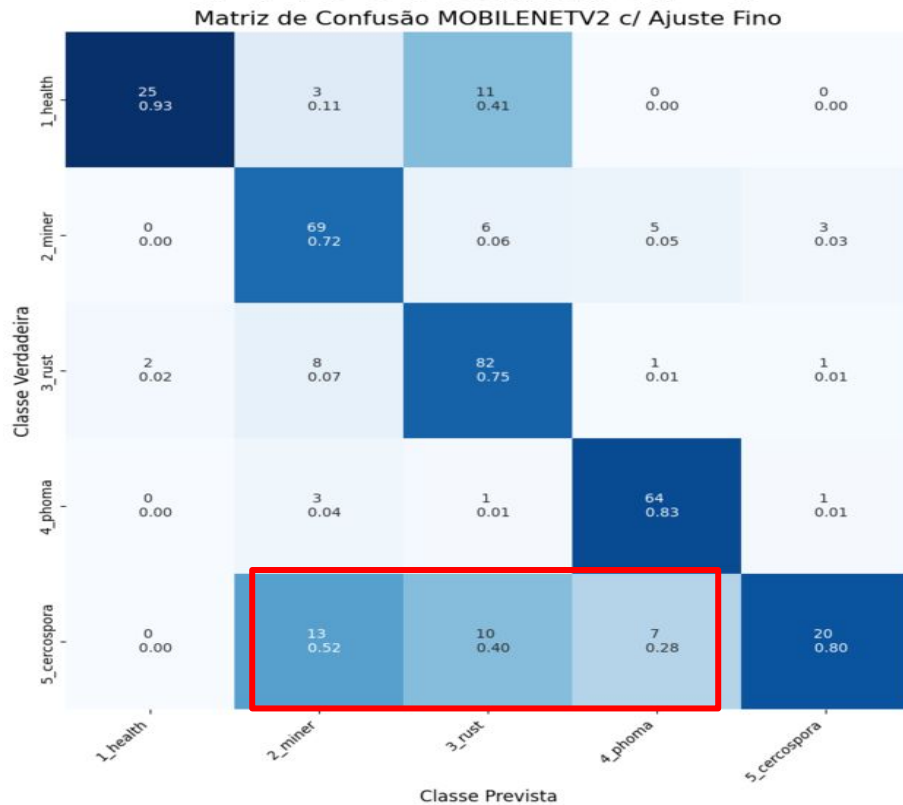
Fonte: Elaborado pelo autor.

Figura 14 - MobileNetV2



Fonte: Elaborado pelo autor.

Figura 15 - MobileNetV2 com ajuste fino



Fonte: Elaborado pelo autor.

5

Conclusões



Conclusões

- Alcançou seu objetivo ao identificar e classificar os estresses bióticos;
- ResNet50V2 apresentou os melhores resultados, com todas as camadas;
- A utilização das técnicas de transferência de aprendizagem melhorou os resultados;
- Iniciou-se a construção de uma base de dados do leste paulista.



Referências

AGRICULTURA, P. e. A. Ministério da. Café no Brasil. 2017. Disponível em: <<https://www.gov.br/agricultura/pt-br/assuntos/politica-agricola/cafe/cafeicultura-brasileira>>.

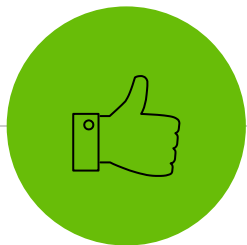
Bezdan, T.; Džakula, N. B. Convolutional Neural Network Layers and Architectures. In: Proceedings of the International Scientific Conference – Sinteza 2019. Novi Sad, Serbia: Singidunum University, 2019. p. 445–451. ISBN 9788679127037. Disponível em: <<http://portal.sinteza.singidunum.ac.rs/paper/700>>.

ESGARIO, J. G.; KROHLING, R. A.; VENTURA, J. A. Deep learning for classification and severity estimation of coffee leaf biotic stress. Computers and Electronics in Agriculture, v. 169, p. 105162, 2020. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169919313225>>.

LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. nature, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. 5, 6

Liu, B.; Zhang, Y.; He, D.; Li, Y. Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry, v. 10, n. 1, 2018. ISSN 2073-8994. Disponível em: <<https://www.mdpi.com/2073-8994/10/1/11>>.

ZHUANG, F.; QI, Z.; DUAN, K.; XI, D.; ZHU, Y.; ZHU, H.; XIONG, H.; HE, Q. A comprehensive survey on transfer learning. Proceedings of the IEEE, v. 109, n. 1, p. 43–76, 2021.



Obrigado!

Perguntas?



*E-mail: luís.hv2306@gmail.com
<https://github.com/LuisHVieira/tcc>*