

# Projeto Inteligência Artificial

Edson Fagner da Silva Cristovam<sup>1</sup>, Erico André Silva<sup>1</sup>, Jadiel Eudes Mendonça Barbosa<sup>1</sup>

<sup>1</sup>Departamento de Computação – Universidade Federal Rural de Pernambuco (UFRPE)  
Rua Dom Manoel de Medeiros, s/n – Campus Dois Irmãos – PE – Brasil

fagner.cristovam@hotmail.com, ericoandresilva@gmail.com, jadielleudes@gmail.com

**Resumo.** *Este artigo descreve um trabalho acadêmico que tem como objetivo utilizar técnicas de aprendizado profundo para treinar uma rede neural que obtenha uma classificação satisfatória de imagens de folhas de fava, mais especificamente, a fava cearense e a fava orelha de vó, que foram abordadas ao longo deste artigo.*

## 1. Introdução

O processo de classificação de folhas, é realizado por profissionais de forma manual, onde demanda uma grande quantidade de tempo, para assim realizá-lo, e a taxa de acerto depende exclusivamente da capacidade de quem está avaliando. Uma alternativa para este problema é a utilização de métodos computacionais de classificação, como Aprendizado Profundo, do inglês Deep Learning, mais especificamente, com a utilização das chamadas, Redes Neurais Convolucionais ou Convolutional Neural Network (CNN).

As Redes Neurais Convolucionais são uma classe de Redes Neurais Artificiais que se mostraram muito eficazes em áreas de reconhecimento de imagens, portanto, na maioria dos casos, são aplicadas ao processamento de imagens. A rede recebe informações, como a representação numérica da imagem, e atribui pesos e desvios efetivamente em seus objetos e, finalmente, é capaz de diferenciar as imagens umas das outras [Sharma 2019].

Por fim, Redes Neurais Artificiais são sistemas paralelos distribuídos compostos por unidades de processamento simples (nodos) que calculam determinadas funções matemáticas (normalmente não-lineares). Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. O funcionamento destas redes é inspirado em uma estrutura física concebida pela natureza: o cérebro humano [de Pádua Braga et al. 2007].

### 1.1. Objetivo

Este trabalho tem como objetivo projetar uma rede neural convolucional para classificar imagens de folhas da Fava Cearense e Orelha de Vó.

## 2. Metodologia

O conjunto de dados utilizado foi disponibilizado por Raphael Miller e possui um total de 229 imagens, sendo 109 de Fava Cearense e 120 de Orelha de Vó. Os arquivos estão disponíveis no site Github

Para realizar a classificação das imagens, foi utilizada Keras, uma API de alto nível de redes neurais, escrita em Python e capaz de executar em cima de TensorFlow, CNTK, ou Theano [Keras.io 2019].

## 2.1. Pré-processamento

Com o intuito de obter um melhor desempenho da rede neural, foi realizado um pré-processamento nas imagens (Figura 1). A partir do conjunto de dados original, foram criados mais três novos conjuntos. O primeiro é composto por imagens que tiveram boa parte do fundo da imagem removido, permanecendo somente as folhas (Figura 2).

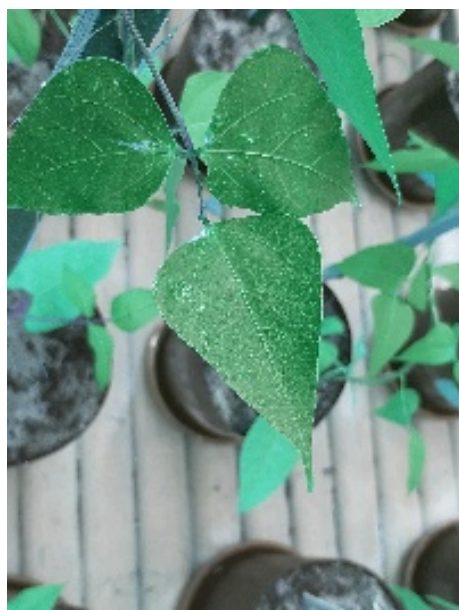


Figura. 1. Com o fundo (original)

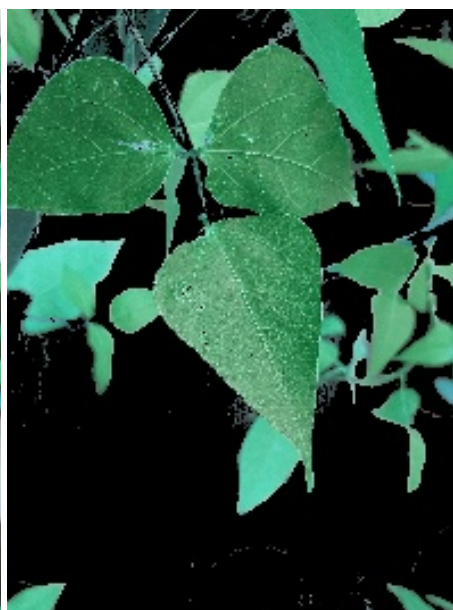


Figura. 2. Sem o fundo

A remoção do fundo foi feita utilizando o OpenCV (Open Source Computer Vision Library), uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto disponível em Python [OpenCV.org 2019]. Foi usado um filtro no padrão HSL (Hue, Saturation, Lightness) também conhecido com HSV (Hue, Saturation, Value), onde as cores são definidas por meio de valores de tonalidade, saturação e brilho. O filtro analisa a imagem pixel a pixel e torna preto todo pixel que não está no intervalo definido. O intervalo de cor usado no filtro foi entre 25 de tonalidade, 52 de saturação e 72 de brilho até 102 de tonalidade, 255 de saturação, 255 e brilho. Com isso os verdes são mantidos, bem como alguns tons de marrom e amarelo que as plantas das imagens possuem.

O segundo conjunto de imagens foi gerado a partir de uma técnica de aumento de dados, onde são geradas novas imagens por meio de algumas modificações como, deslocamento, mudança do brilho, rotação, entre outros. As novas imagens foram criadas utilizando modificações aleatórias de espelhamento horizontal, espelhamento vertical ou variação do brilho em 50% para mais ou para menos. Para cada imagem, foram gerada cinco imagens modificadas, usando de uma a todas as modificações supracitadas. Após a aplicação desta técnica, o conjunto de dados aumentou para 654 imagem de Fava Cearense e 720 imagens de Orelha de Vó, totalizando 1374 imagens (Figura 3 e Figura 4).



Figura. 3. Imagem modificada 1      Figura. 4. Imagem modificada 2

O terceiro conjunto de imagens foi criado usando uma combinação das duas técnicas anteriores. Uma visão geral da distribuição dos conjuntos de imagens pode ser vista na (Tabela.1).

Após a criação dos novos conjuntos de imagens, foi aplicado um redimensionamento em todos os conjuntos usando a biblioteca OpenCV, com a finalidade de reduzir o tamanho das imagens de 3456x4608 pixel para 200x267 pixel, mantendo a mesma proporção. Esse passo foi incluído devido ao tempo limitado para execução dos teste, bem como a limitação de hardware para contas gratuitas do Google Colab, plataforma usada para executar os algoritmos de treinamento da rede [Google 2019].

<b>Conjunto</b>	<b>Fava Cearense</b>	<b>Orelha de Vó</b>	<b>Total</b>
Imagens Normais	109	120	229
Imagens Sem Fundo	109	120	229
Imagens Editadas	654	720	1374
Imagens Editadas e Sem Fundo	654	720	1374

Tabela. 1. Distribuição dos conjuntos de imagens após o pré-processamento

## 2.2. Treinamento

Os treinamentos foram feitos utilizando cada um dos quatro conjuntos de imagens gerados a partir do original. Eles foram divididos em 60% para treinamento e 40% para teste. As imagens foram selecionadas aleatoriamente a cada treino realizado. O treinamento foi realizado com 80 épocas, este valor foi escolhido, pois em teste preliminares, foi o número de épocas que trouxe acurácia mais satisfatória, havendo queda na taxa de acerto para épocas menores e muito maiores ao usar pesos aleatórios, como pode ser visto na (Figura 5). Esses teste preliminares foram realizados somente no conjunto original

redimensionado para 200x267 pixel. O otimizador escolhido foi Adam "gradiente descendente estocástico". O escolhemos por trazer bons resultados em trabalhos anteriores.

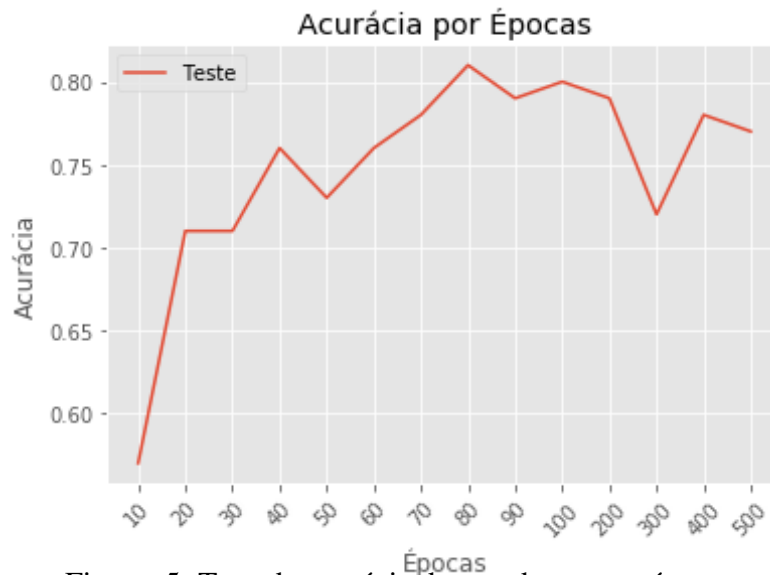


Figura. 5. Taxa de acurácia de acordo com as épocas

Detalhes da arquitetura da rede podem ser observados na (Figura 10). Ela foi usada para treinar os quatro conjuntos de imagens. Cada conjunto foi treinado duas vezes, a primeira usando o padrão de cor RGB, já a segunda vez, o padrão de cor usado foi a escala de cinza. Após o treinamento, foram selecionados o modelo com melhor acurácia treinado com imagens em RGB e o melhor modelo treinado com imagens em escala de cinza e salvo os seus respectivos pesos. O processo de treinamento foi repetido mais uma vez, porém utilizando os pesos salvos.

Como citado anteriormente, podemos perceber nas (Figura 6) e (Figura 7) a acurácia obtida com redes treinadas sem e com pesos definidos, na qual as imagens estão no padrão RGB.

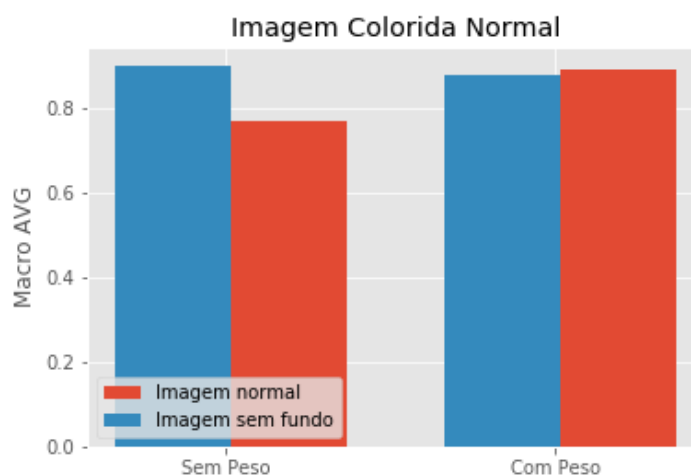


Figura. 6. Taxa de acerto dos modelos treinados com e sem pesos definidos, usando imagens RGB e 200x267 pixel

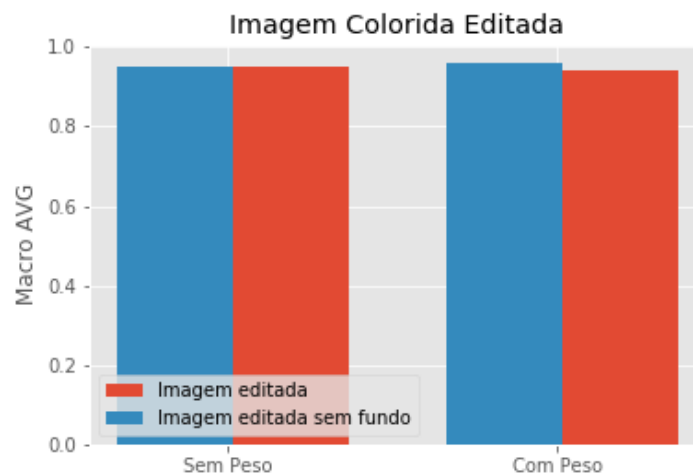


Figura. 7. Taxa de acerto dos modelos treinados com e sem pesos definidos, usando imagens RGB, editadas e 200x267 pixel

Da mesma forma foi feito com as imagens em escala de cinza e pode ser observado a diferença de acurácia nas (Figura 8) e (Figura 9).

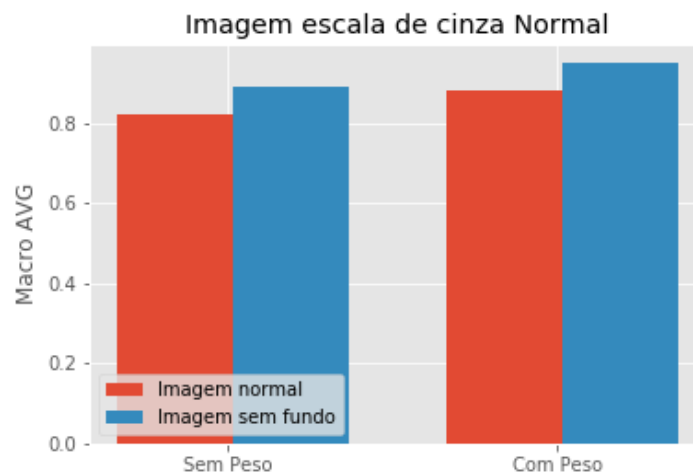


Figura. 8. Taxa de acerto dos modelos treinados com e sem pesos definidos, usando imagens em escala de cinza e 200x267 pixel

### 3. Resultados

Ao realizar o treinamento das quatro bases geradas, como foi explicado na Seção 2.2. Treinamento, utilizando o padrão de cores RGB, foi observado bons valores de acurácia, tendo um valor mais significativo o modelo treinado com a base editada onde as imagens não tem fundo, como pode ser visto na (Figura 11).

Entretanto, ao realizar o treinamento nas mesmas bases, mas mudando o padrão de cor para escala de cinza, foi obtido uma acurácia maior que as dos modelos treinados com imagens RGB, onde o modelo treinado com as imagens editadas e sem fundo, obteve uma acurácia de 0.99, como pode ser observado na (Figura 12).

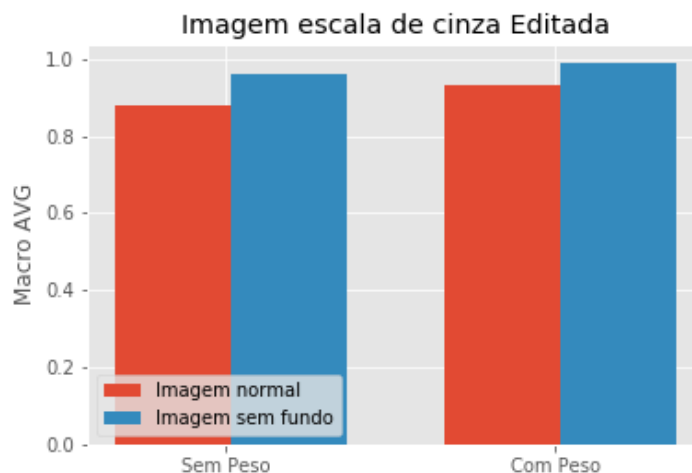


Figura. 9. Taxa de acerto dos modelos treinados com e sem pesos definidos, usando imagens em escala de cinza, editadas e 200x267 pixel

#### 4. Dificuldades

Uma das dificuldades encontradas nesse trabalho, se dá pela limitação tanto de memória RAM quanto da GPU para treinar a rede com imagens grandes. Os modelos foram treinados utilizando uma conta gratuita no Google Colab que tem um hardware mais apropriado que os disponíveis para a realização deste trabalho, no entanto, por ser uma conta gratuita, esse poder computacional ainda é limitado para processar o conjunto de dados original. Devido a isso, foi necessário a redução do tamanho das imagens, antes de treinar a rede, para um tamanho de 200x267 pixel. Esse tamanho mantém a proporção das imagens originais. Os resultados obtidos poderiam ser melhores caso a rede fosse treinada com as imagens no tamanho original. Outro fator que poderia melhorar os resultados obtidos, seria um número maior de imagens para o treinamento da rede.

A maneira como as imagens foram fotografadas não é a mais indicada, pois tudo que não é o objeto da classificação, como chão, vasos e sujeira, gera um ruído no aprendizado. No entanto, a inclusão desses elementos na imagem pode ajudar a rede na classificação correta de imagens em ambientes diversos.

O tempo para a execução do projeto também se tornou um problema, pois o contato com a base de dados foi tardio, com isso, menos testes foram realizados para conclusão deste trabalho no prazo determinado.

#### 5. Conclusão

É notório a eficiência de Redes Neurais Convolucionais para a classificação de imagens, que obteve um ótimo valor de acurácia. Com a rede treinada há um ganho muito grande de tempo para classificar uma folha em relação ao especialista.

Contudo, ainda há melhorias a serem feitas para treinar essa rede, como imagens mais adequadas no conjunto de dados, assim como uma maior quantidade e diversidade de folhas como também de suas respectivas classes, a fim de gerar valores satisfatórios com os diversos tamanhos de imagens.

## References

- [de Pádua Braga et al. 2007] de Pádua Braga, A., de Leon Ferreira, A. C. P., and Ludermit, T. B. (2007). *Redes neurais artificiais: teoria e aplicações*. LTC Editora Rio de Janeiro, Brazil:.
- [Google 2019] Google (2019). Google colabory. <https://colab.research.google.com/>. Accessed: 2019-11-18.
- [Keras.io 2019] Keras.io (2019). Keras: The python deep learning library. <https://keras.io/>. Accessed: 2019-11-18.
- [OpenCV.org 2019] OpenCV.org (2019). Opencv. <https://opencv.org/about/>. Accessed: 2019-11-18.
- [Sharma 2019] Sharma, V. (2019). Deep learning – introduction to convolutional neural networks — vinod sharma’s blog. <https://vinodsblog.com/2018/10/15/everything-you-need-to-know-about-convolutional-neural-networks/>. Accessed: 2019-11-18.

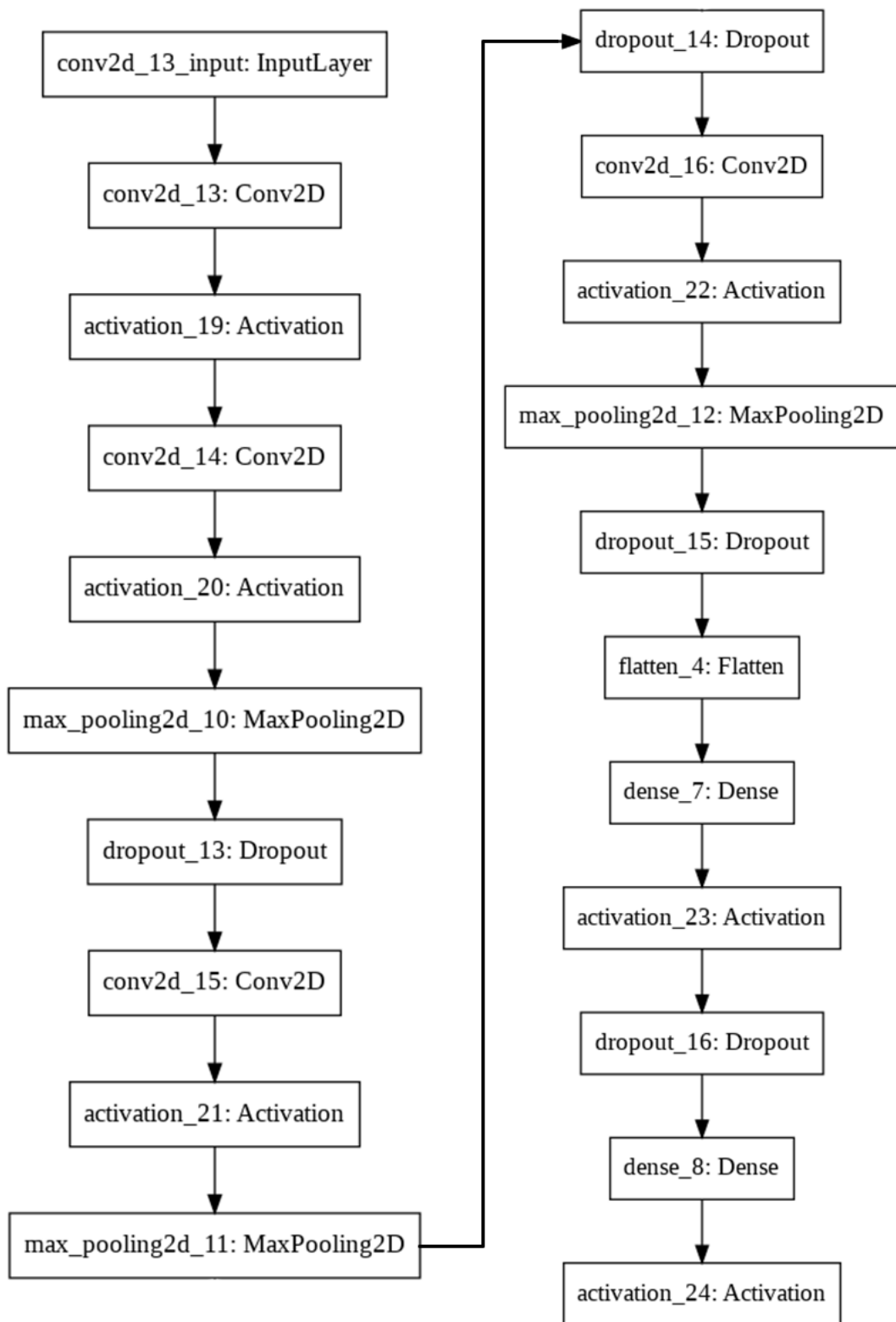


Figura. 10. Arquitetura da rede



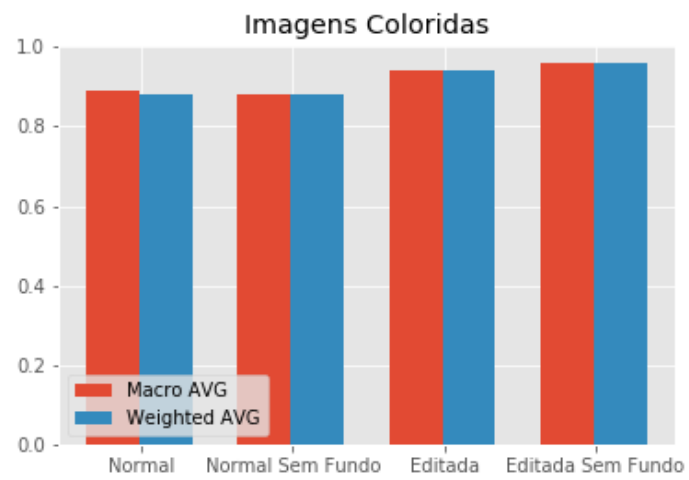


Figura. 11. Taxa de acerto dos modelos treinados com imagens (RGB)

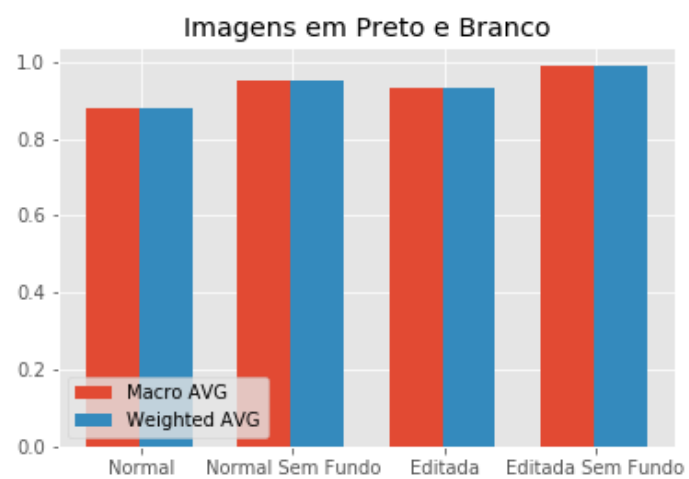


Figura. 12. Taxa de acerto dos modelos treinados com imagens em escala de cinza