Classificação de folhas de plantas utilizando CNN

Alex Seródio Gonçalves e Luma Kühl
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil
{alexserodio,lkuhl}@furb.br

1 Introdução

Este relatório tem como objetivo descrever o desenvolvimento do trabalho final da disciplina de Processamento de Imagens. O qual foi desenvolvido com o propósito de reconhecer folhas de plantas através de redes neurais convolucionais fazendo uso do dataset FLAVIA criado por Stephen Gang Wu*, et al*, em seu paper entitulado *A Leaf Recognition Algorithm for Plant classification Using Probabilistic Neural Network* [1].

Neste trabalho descreveremos as técnicas utilizadas para a realização do processo de classificação, como também do tratamento do dataset escolhido. Apesar do dataset FLAVIA contar com 33 espécies de folhas diferentes, para o nosso desenvolvimento selecionamos apenas 10 espécies. Para o desenvolvimento no algoritmo utilizamos a linguagem de programação Python junto com as bibliotecas *TensorFlow* e *Keras*.

2 Desenvolvimento

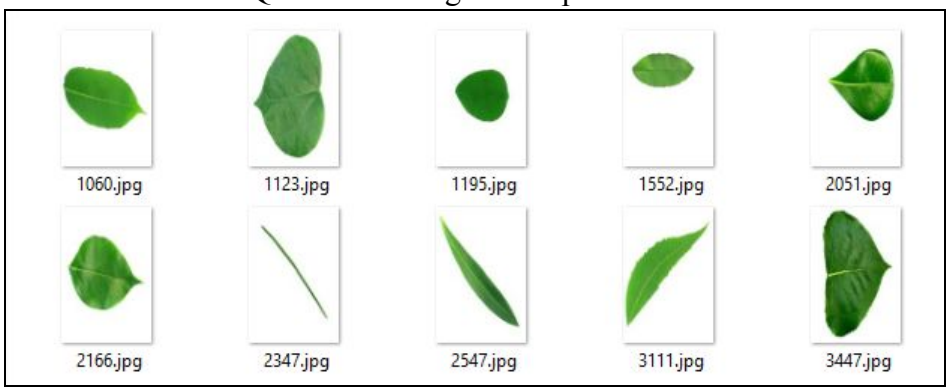
A seguir serão demonstrados os processos para preparação do dataset como também do modelo final obtido e utilizado neste trabalho.

2.1 Preparação do Dataset

O FLAVIA dataset possui um total de 1907 imagens distribuídas em 33 espécies de plantas. Em média cada planta possui cerca de 56 imagens que devem ser divididas para treinamento e teste.

Como não havia necessidade de se trabalhar com todas as 33 espécies de plantas contidas no dataset original, selecionamos a partir delas apenas as 10 espécies que possuíam a maior quantidade de imagens. Após essa extração, nosso dataset ficou com um total de 610 imagens. No Quadro 1 é possível observar as imagens das espécie selecionada.

Quadro 1 - Imagens das plantas do dataset.



Fonte: elaborado pelos autores.

Após a extração, ainda foi necessário diminuir o tamanho das imagens, visto que cada imagem possuia dimensões de 1600 x 1200. Esse tamanho foi reduzido para 150 x 100, o que facilita muito o processamento das mesmas. Além da redução do tamanho, todas as imagens também foram convertidas em escala de cinza.

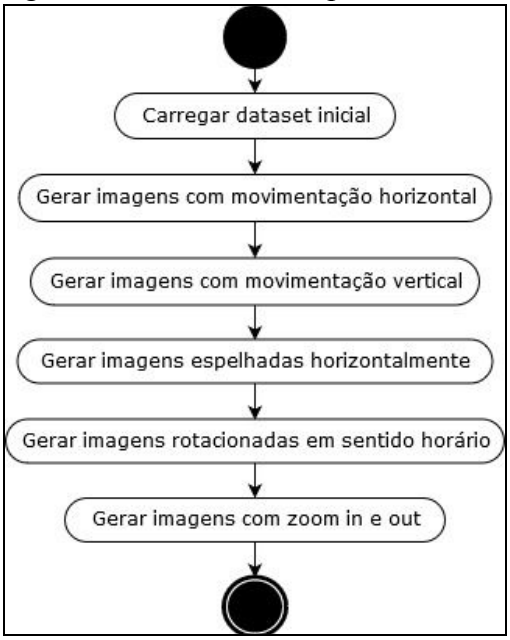
Este dataset gerado foi então separado em 90% para treino e 10% para teste e submetido ao nosso modelo de treinamento. Porém, ao executar o treinamento e testes várias vezes obtinhamos sempre 100% de acurácia, o que indicava claramente um caso de *overfitting*, muito provavelmente devido ao fato de nosso dataset ser muito pequeno para ser utilizado por uma rede convolucional. Foi necessário então realizar um procedimento para a ampliação do dataset.

2.2 Data Augmentation

Data augmentation é o nome dado ao processo que tem como objetivo expandir a quantidade de imagens de um dataset através da geração de novas imagens a partir das imagens já existentes. Neste caso, optamos por realizar a expansão do nosso dataset através do processo descrito por Jason Brownlee em seu artigo *How to Configure Image Data Augmentation when Training Deep Learning Neural Networks* [2].

A Figura 1 apresenta o diagrama de atividades dos processos realizados pelo método para fazer o data augmentation a partir de um conjunto de imagens de entrada. Cada etapa descrita no diagrama (com exceção da primeira) é responsável por gerar nove novas imagens com pequenas diferenças aleatórias entre elas alterando levemente o fator de transformação utilizado.

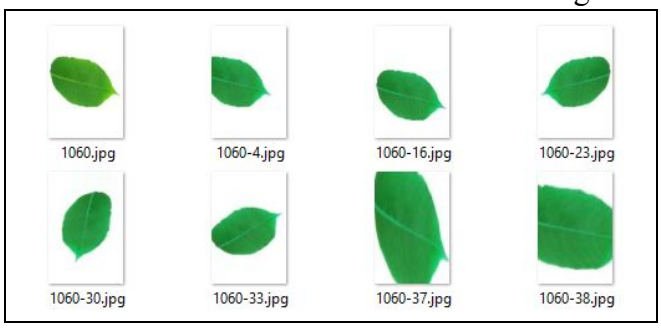
Figura 1: Diagrama de atividades do processo de Data Augmentation.



Fonte: elaborado pelos autores.

Ao final do processo são geradas 45 novas imagens a partir de cada imagem existente no dataset inicial, o que resulta em um novo dataset com um total de 28.123 imagens, sendo que dessas, 80% foram utilizadas para treino e os outros 20% para teste.

Quadro 2 - Resultados obtidos através do data augmentation



Fonte: elaborado pelos autores.

Além disso, a partir do dataset ampliado também foram gerados dois novos datasets, um com todas as imagens binarizadas e outro com todas as imagens em escala de cinza, totalizando três datasets diferentes que serão utilizados na validação do modelo.

2.3 Modelo

O modelo utilizado é formado por três camadas convolucionais compostas por 16, 32 e 64 neurônios, respectivamente. Todas utilizam a função de ativação *ReLu* e são intercaladas por camadas de *Max Pooling*. Por fim, como estamos trabalhando com dez possibilidades o modelo termina com uma camada *Dense* com 10 neurônios e a função de ativação *softmax*.

```
01 model = keras.Sequential([
02       keras.layers.Conv2D(16, 5, padding='same', activation='relu',
   input_shape=(dataset.height, dataset.width, 3)),
03       keras.layers.MaxPooling2D(),
04       keras.layers.Conv2D(32, 5, padding='same', activation='relu'),
05       keras.layers.MaxPooling2D(),
06       keras.layers.Conv2D(64, 5, padding='same', activation='relu'),
07       keras.layers.MaxPooling2D(),
08       keras.layers.Flatten(),
09       keras.layers.Dense(10, activation='softmax')
10 ])
```

Fonte: elaborado pelos autores.

3 Resultados e discussões

Conforme pode ser observado no Quadro 5, a acurácia da rede se manteve relativamente próxima com os três tipos de dataset, tendo uma assertividade de cerca de 90% ao ser validada com os dados de teste após 10 sessões de treino.

Quadro 5 - Resultados obtidos com diferentes formatos de dataset

dataset colorido

Época	Acurácia
1	0.7728
2	0.9278
3	0.9500
4	0.9521
5	0.9628
6	0.9615
7	0.9564
8	0.9764
9	0.9706
10	0.9734

Acurácia do teste: 0.8730

dataset grayscale

Época	Acurácia
1	0.7578
2	0.9179
3	0.9369
4	0.9469
5	0.9587
6	0.9617
7	0.9641
8	0.9661
9	0.9731
10	0.9759

Acurácia do teste: 0.9365

dataset binarizado

Época	Acurácia
1	0.7434
2	0.8838
3	0.9182
4	0.9138
5	0.9478
6	0.9590
7	0.9669
8	0.9695
9	0.9745
10	0.9737

Acurácia do teste: 0.9157

Fonte: elaborado pelos autores.

O uso de data augmentation se mostrou uma solução efetiva para o problema de *overfitting* encontrado com o dataset original. Porém, visto que o processo de data augmentation utilizado gera imagens relativamente semelhantes e nossas imagens iniciais de cada espécie já não eram tão diferentes entre si ainda existe a possibilidade de nosso dataset estar enviesado, e resta a dúvida de se um dataset diferente e maior que o nosso inicial teria um resultado melhor.

4 Conclusões

Uma acurácia de aproximadamente 90% (93.65% no melhor caso com o dataset em escala de cinza) é satisfatória dado o objetivo inicial proposto. A maior dificuldade encontrada foi definitivamente a dificuldade em encontrar um dataset adequado ao cenário proposto. Como um dataset verdadeiramente adequado não foi encontrado, foram necessárias algumas manipulações sobre ele para ser possível utilizar um modelo de rede convolucional. Os processos escolhidos para a correção do dataset foram efetivos mas demandaram de boa parte do tempo de desenvolvimento deste trabalho.

5 Referências

[1] *A Leaf Recognition Algorithm for Plant classification Using Probabilistic Neural Network,* Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang and Chiao-Liang Shiang. *IEEE 7th International Symposium on Signal Processing and Information Technology, Dez. 2007, Cario, Egito*

[2] *How to Configure Image Data Augmentation when Training Deep Learning Neural Networks,* Jason Brownlee. Disponível em: https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/