

tratamento de imagens

michelvictor

May 2019

1 Base Ambrapa

PDDb possui 2326 imagens de 171 doenças e outras desordens acometendo 21 espécies de plantas [1]. Apesar de sua significância, PDDb não possui representatividade suficiente para permitir o uso de técnicas como “Deep Learning”. Então foi usada a base XDB, que foi criada a partir da PDDb, seguidos alguns critérios [1]. Como resultado, esta base de imagens expandida (XDB) atualmente contém 46.513 imagens [1].

2 Tratamento da base XDB (Cropped)

Para agilizar o processo foi criado um script para automatizar o processo de tratamento da base. O tratamento da imagem possui dois processos principais, o primeiro é validar a imagem, caso em que ela não possuir tamanho suficiente para ser usada, segundo redimensionar todas as imagens para uma mesma dimensão.

Para rodar o script[2] é necessário adotar uma arquitetura de pastas

2.1 Validação de Imagem

Um dos problemas da base XDB é que ela possui imagens muito pequenas, sem nenhuma representatividade Figura 1. Para resolver esse problema foi implementado um filtro:

```
1 # Funcao de validacao
2 # num_px e num_py, sao as novas dimensoes da imagem
3 def validaImagem(imagem:Image, taxaDeDiferenca=None)-> bool:
4     width, height = 0, 0
5     if taxaDeDiferenca:
6         width, height = (taxaDeDiferenca * num_px,
7                           taxaDeDiferenca * num_py)
8     if imagem.size[1] >= (num_px - width) or imagem.size[0]
9         >= (num_py - height):
10         return True
11     return False
```

Considerando *taxaDeDiferenca=None*, o que estar sendo feito na **linha 9** é uma verificação simples entre o tamanho da imagem com o novo tamanho que ela terá (*num_px, num_py*), se a dimensão da imagem (tupla *imagem.size*) for menor, a mesma é descartada. Caso a função receba *taxaDeDiferenca*, que é um valor em porcentagem, imagens com o tamanho menor que *num_px* e *num_py*, podem ser usado, caso sejam maior que a diferença entre a porcentagem (*taxaDeDiferenca*) em relação a *num_px* e *num_py* com (*num_px, num_py*)



Figure 1: 17x19 pixels

2.2 Redimensionamento das Imagens

O Redimensionamento das imagens é bem simples, o segundo *for* serve para abrir cada imagem de uma determinada classe, e em seguida na **linha 4** a chamada do função de validação, caso em que a imagem não esteja (*False*) adequada, pula para próxima imagem **linha 7 e 8**, e no caso contrario (*True*), a imagem vai ser redimensionada e em seguida salva.

```

1 for ... # for para cada classe/pasta
2     for pathImg in imagens:
3         image = Image.open(pathImg)
4         if validaImagem(image, TAXA_DIFERENCA):
5             newImage = image.resize((num_px, num_py))
6             newImage.save(NEW_BASE + '/' + labels[count] + '-'
7                             + str(count) + '/' + str(countImg) + EXTESAO_IMG )
8         else:
9             continue

```

O resultado para uma imagem é mostrado nas Figuras 2 e 3

References

- [1] B. A. Halfeld-Vieira et al. J. G. A. Barbedo, L. V. Koenigkan. Annotated plant pathology databases for image-based detection and recognition of diseases. *IEEE LATIN AMERICA TRANSACTIONS*, 16:1749–1757, 2018.
- [2] Michel Victor. Algoritmo para o tratamento (redimensionamento) de imagem para uso em deep learning., may 2019. <https://github.com/michvic/tratamento-imagem-deep-learning>.



Figure 2: Antes do processamento



Figure 3: Depois do processamento