
PYCON
AMAZÔNIA

Aprendizado de máquina na classificação de plantas amazônicas

Uma introdução ao projeto W.A.V.A.

Allex Lima e Marcos Salame

12 de agosto de 2017

Embrapa Amazônia Ocidental

Outline

Um pouco sobre a Embrapa


Missão
Viabilizar soluções de pesquisa, desenvolvimento e inovação para a sustentabilidade da agricultura, em benefício da sociedade brasileira.
Visão
Ser referência mundial na geração e oferta de informações, conhecimentos e tecnologias, contribuindo para a inovação e a sustentabilidade da agricultura e a segurança alimentar

17
Unidades Centrais localizadas em Brasília

46
Unidades Descentralizadas em todas as regiões do Brasil

4
Laboratórios Virtuais no Exterior (Labex), nos EUA, Europa, China e Coreia do Sul

3
Escritórios Internacionais na América Latina e África

Algumas unidades

- **Embrapa Informática Agropecuária**
  - Localizada em Campinas - SP
  - Atua com engenharia de sistemas de software, computação científica, tecnologia de comunicação, bioinformática e agroclimatologia
- **Embrapa Instrumentação**
  - Localizada em São Carlos - SP
  - Trabalha em áreas como física e engenharia
- **Embrapa Monitoramento por Satélite**
  - Localizada em Campinas - SP
  - Opera em áreas de inovações geoespaciais

Plantas Amazônicas

A Amazônia

contém mais de 30.000 espécies

10% das plantas de todo o planeta

Fonte: Museu Emílio Goeldi


O Guaraná

- O fruto do guaranazeiro (*Paullinia cupana* var. *sorbilis*), o **guaraná**, é utilizado como um dos **principais insumos em indústrias de bebidas e cosméticos**;
- Entretanto, **a proliferação de doenças e pragas incita uma baixa produtividade** nos estados que o cultivam de forma tradicional, a partir **mudas oriundas de sementes**;
- A Embrapa, então, tem disponibilizado **19 cultivares resistentes a diversos fatores bióticos e abióticos** que afetam a guaranaicultura na região amazônica, desde a década de 70.

Contribuição ao setor agronômico; enriquecimento da literatura computacional.

- iNaturalist
- Tela Botanica
- ImageClef
  - Pl@ntNet

O projeto WAVA

- **Identificar** os principais **cultivares de guaranazeiros**
  - A partir de uma **amostra foliar**
  - Utilizando técnicas de **aprendizado de máquinas**
    - DT + HOG
    - SVM (*Kernel* RBF) + HOG
    - CNN*
- Visando auxiliar o trabalho de técnicos, pesquisadores e guaranaicultores

Extração de características

H*istogram of* **O***riented* **G***radients* – HOG

(Histograma de gradientes orientados)

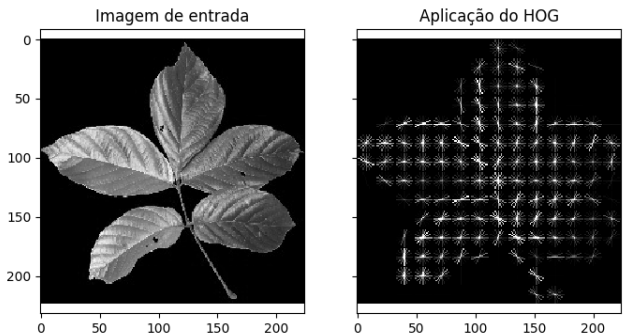


Figura 1: Entrada e resultado do processo de extração de características, através do método HOG.

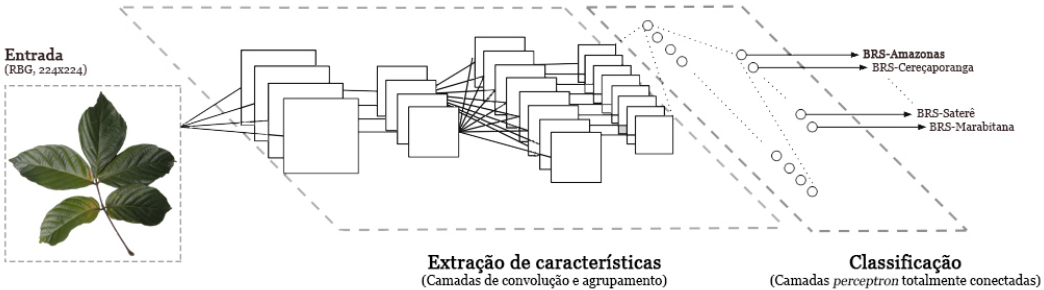


Figura 2: Ilustração do processo de classificação dos cultivares de guaraná através da arquitetura LeNet, um modelo de CNN.

O Dataset

- Como ainda **não existe imagens das folhas dos cultivares** de guaranazeiros em **bases de imagens públicas**, como o *ImageNet*, *P@ntNet* e *Flavia dataset*
- Foi essencial o **desenvolvimento de um dataset próprio**
  - Dois cultivares: **BRS-Amazonas** e **BRS-Cereçaporanga**
  - *data augmentation*

Pré-processamento de imagens

a)

b)

Figura 3: Amostras do *dataset* após pré-processamento e *data augmentation*. Na primeira linha, variedades do cultivar BRS-Amazonas e, na linha b), do BRS-Cereçaporanga.

Performance de classificação

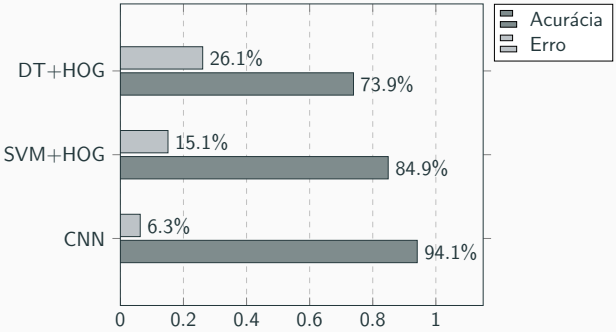
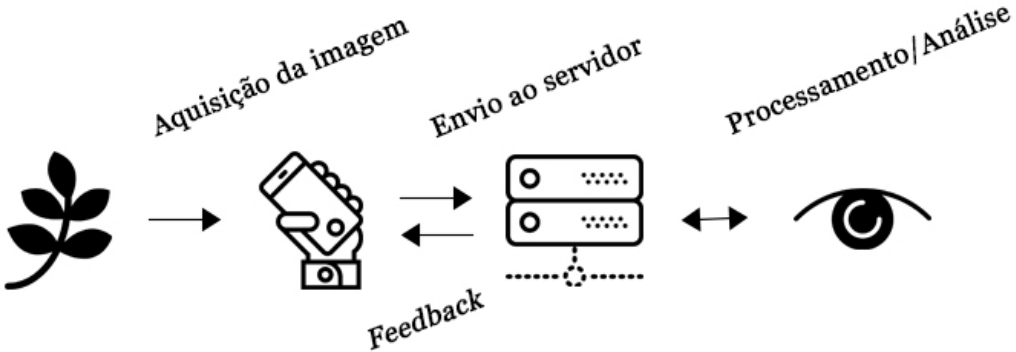
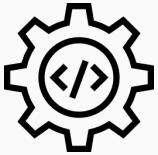


Figura 4: Performance média das técnicas de classificação utilizadas

Arquitetura do projeto



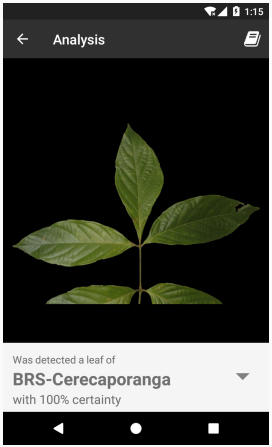
Ambientes



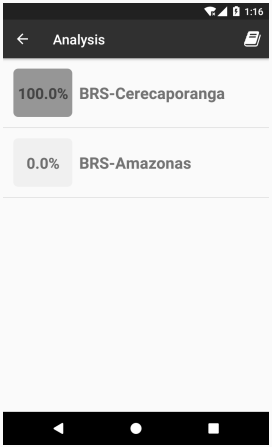
- Servidor/Classificador: **Python 3.x**
  - IDE *Spyder* (*Scientific PYthon Development EnviRonment*);
  - Bibliotecas *Flask*, *Scikit-learn*, *Scikit-image*, *Keras*, *NumPy* e *Matplotlib*
  - Treinamento e homologação:
    - Instância dedicada do **Google Cloud Platform** com processamento *Intel® Xeon®* de 16 núcleos com 2.6*GHz* cada, 30*GBi* de memória *RAM* e sistema operacional *GNU/Linux Debian*.
- Cliente: **Java 8.x** p/ Google Android OS
  - IDE Google Android Studio v2.x; e
  - Android SDK v16 (*Jelly Bean* 4.0+);

Client: Google Android OS App

(a) "Tela inicial"



(b) Feedback



(c) Detalhes

Algumas ferramentas úteis


TensorFlow


scikit
learn


NumPy


spyder

Uma CNN em minutos

```
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3
4 from keras.models import Sequential
5 from keras.layers.convolutional import Conv2D, MaxPooling2D
6 from keras.layers.core import Flatten, Dense, Dropout
7 from keras.preprocessing.image import ImageDataGenerator
8 from keras.optimizers import SGD
```

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from keras.models import Sequential
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.layers.core import Flatten, Dense, Dropout
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import SGD

def lenet(height, width, depth, class_n):
    model = Sequential()
    model.add(Conv2D(10, (5, 5), strides=(1, 1), activation="relu",
                     padding="same", input_shape=(height, width, depth)))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(25, (5, 5), strides=(1, 1), activation="relu",
                     padding="same"))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
    model.add(Dense(300, activation="relu"))
    model.add(Dropout(0.4))

    model.add(Dense(class_n, activation="sigmoid"))

    return model
```

```
29 if __name__ == "__main__":
30     width, height, channels = 124, 124, 3
31     batch_size = 50
32 
33     datagen = ImageDataGenerator(
34         rotation_range=90,
35         width_shift_range=0.01,
36         height_shift_range=0.01,
37         zoom_range=0.05,
38         channel_shift_range=0.05,
39         horizontal_flip=True,
40         vertical_flip=True,
41         fill_mode='constant',
42         cval=0)
```

```
29 if __name__ == "__main__":
30     width, height, channels = 124, 124, 3
31     batch_size = 50
32 
33     datagen = ImageDataGenerator(
34         rotation_range=90,
35         width_shift_range=0.01,
36         height_shift_range=0.01,
37         zoom_range=0.05,
38         channel_shift_range=0.05,
39         horizontal_flip=True,
40         vertical_flip=True,
41         fill_mode='constant',
42         cval=0)
44     train_generator = datagen.flow_from_directory(
45         'dataset/train',
46         target_size=(height, width),
47         batch_size=batch_size)
48 
49     validation_generator = datagen.flow_from_directory(
50         'dataset/validation',
51         target_size=(height, width),
52         batch_size=batch_size)
```

```
54      opt = SGD(lr=0.01)
55      cnn = lenet(height, width, channels, 3)
56      cnn.compile(loss="categorical_crossentropy", optimizer=opt,
57                  metrics=["accuracy"])
58      cnn.fit_generator(
59              train_generator,
60              steps_per_epoch=150,
61              epochs=2,
62              validation_data=validation_generator,
63              validation_steps=150)
64
65      cnn.save_weights('my-cnn-weights.h5')
```

```
54      opt = SGD(lr=0.01)
55      cnn = lenet(height, width, channels, 3)
56      cnn.compile(loss="categorical_crossentropy", optimizer=opt,
57                  metrics=["accuracy"])
58      cnn.fit_generator(
59              train_generator,
60              steps_per_epoch=150,
61              epochs=2,
62              validation_data=validation_generator,
63              validation_steps=150)
64
65      cnn.save_weights('my-cnn-weights.h5')
```

Disponível em: `github.com/allexlima/pycon-lenet`


That's all Folks!

PyNorte

Telegram: `bit.ly/pynorte`
Website: `pynorte.python.org.br`
Github: `github.com/pynorte`
Facebook: `facebook.com/pynorte`

Obrigado

allexlima@unn.edu.br
allexlima.com
marcos.salame@embrapa.br
github.com/mfas

compiled in LaTeX
using Metropolis theme

(github.com/matze/mtheme)