

Bean photo classification

Task 1

Design a classifier that performs multiclass classification of healthy i
sick beans:

<https://www.tensorflow.org/datasets/catalog/beans>

<https://github.com/AI-Lab-Makerere/ibean/>

The data is taken from the server in the form of a tensorflow dataset. API description:

<https://www.tensorflow.org/guide/data>

https://www.tensorflow.org/api_docs/python/tf/data/Dataset

Tip 1:

- ds.map(): data preprocessing and augmentation
- ds.shuffle(): shuffle data before each epoch or iteration
- ds.batch(): batch size setting
- ds.repeat(): repeating examples from a dataset after analyzing all of them

Elements

Tip 2:

Training examples should be mixed at least before each epoch.

GPU can be used to train the network if available. For this purpose go to *Runtime* -> *Change runtime type* and select *GPU as Hardware accelerator*.

You can get started with the following code:

```
!pip install -U tensorflow_datasets  
  
import tensorflow as tf  
  
import tensorflow_datasets as tfds  
  
# Load train, validation and test datasets  
  
train, train_info = tfds.load(name='beans', split='train', shuffle_file  
s=True, with_info=True)
```

```
valid, valid_info = tfds.load(name='beans',split='validation', shuffle_
files=False, with_info=True)

test, test_info = tfds.load(name='beans', split='test', shuffle_files=F
alse, with_info=True)

# Show examples of images and corresponding classes

tfds.show_examples(test, test_info)

# Print example image shape and label

example_dict = next(iter(train))

print('Image shape', example_dict['image'].shape)

print('Label', example_dict['label'])
```

Recommended treatments to test:

- reducing the size of images to speed up training
- augmentation of training images, e.g. by rotations, mirror images, slight noise, brightness change, contrast change
- normalization to the range $<-0.5, 0.5>$, $<-1, 1>$ or $<0, 1>$
- BatchNormalization after convolutional and dense layers with large learning rate (0.01-0.1)
- classifier architecture modeled on VGG or Resnet models
- (alternatively) transfer learning and fine tuning with the VGG model or Resnet