Sign Detection

Updated: January 29, 2020

Started 1/28/2020

Keras Demo

Pyenv

As of now, Tensorflow requires python 3.7, and the current python version is 3.8. To provide python 3.7 to this project only, I used `pyenv` (`https://github.com/pyenv/pyenv`). Since I'm using Arch Linux, I used `pacman` to install pyenv, `sudo pacman -S pyenv`. Then I added the following lines to my bash_profile:

```
export PYENV_ROOT="$HOME/.pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"

if command -v pyenv 1>/dev/null 2>&1; then
    eval "$(pyenv init -)"
fi
```

Then I restarted my shell and went to my project directory and ran `pyenv local 3.7.6`.

In this directory I installed packages, via pip, for this version. For example, `sudo pip install matplotlib`. Refer to the github link above for more information.

Emacs Pyenv Mode

Emacs has the `pyenv-mode-auto` package which, when installed, will automatically detect the project's python version (via the .python-version file generated with `pyenv local 3.7.6` above). This means that we can run org-mode python code blocks will use the python version detected, instead of the system python version (which is currently python 3.8).

Data Preperation

Downloading Images from AWS S3 Bucket

```
if [[ ! -d "images" ]]; then
    wget https://s3.us-east-2.amazonaws.com/naturalimages02/images.tar.gz
    tar -xzf images.tar.gz
    rm -f images.tar.gz*
fi;
echo "DONE!"
```

Preparing Training Data

```
from keras.preprocessing.image import ImageDataGenerator
import numpy as np

train_path = '../images/train/'
test_path = '../images/test/'
batch_size = 16
image_size = 224
num_class = 8

train_datagen = ImageDataGenerator(
    validation_split = 0.3,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True)

train_generator = train_datagen.flow_from_directory(
    directory = train_path,
    target_size = (image_size,image_size),
    batch_size = batch_size,
    class_mode = 'categorical',
    color_mode = 'rgb',
    shuffle = True)

x_batch, y_batch = train_generator.next()
```

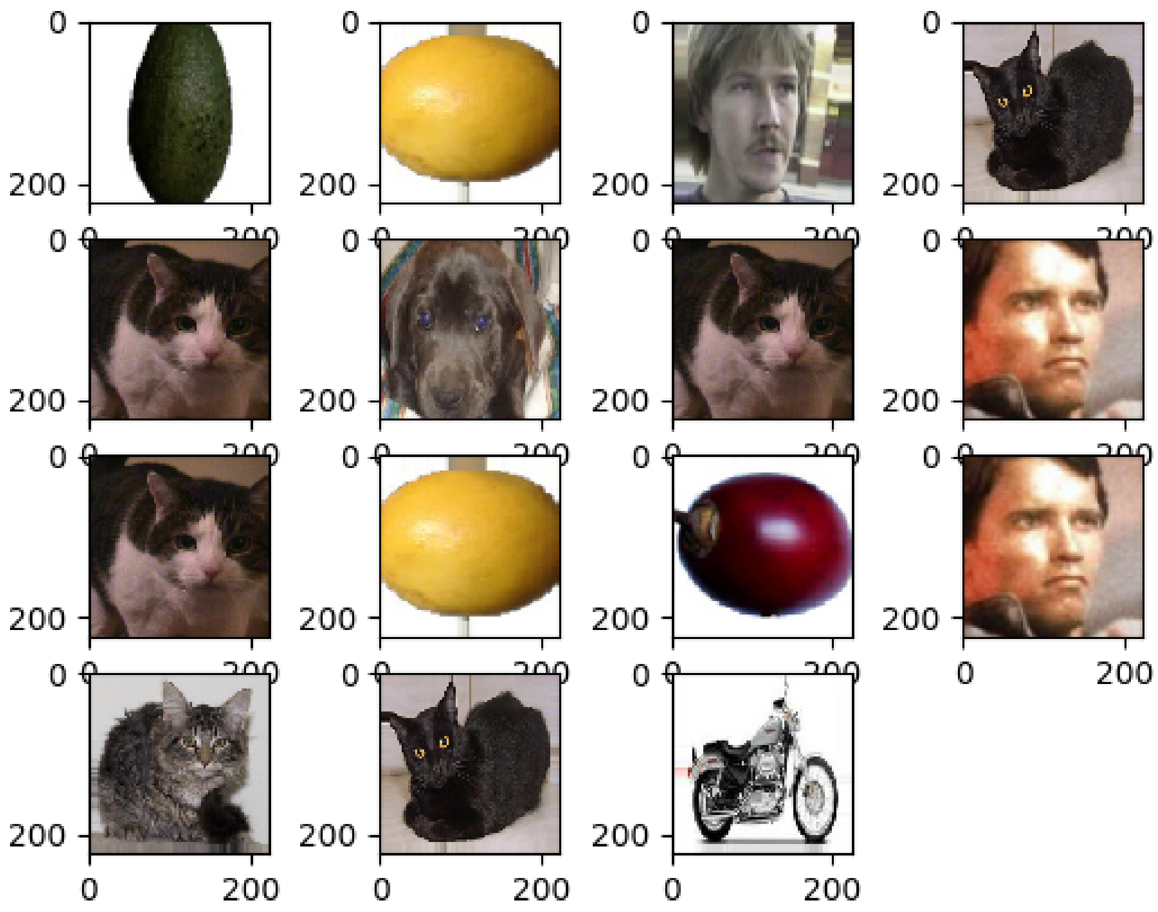
Let's See the Batch

```
import matplotlib.pyplot as pyplot
import numpy
import batchPreparation as bp

fig=pyplot.figure()
columns = 4
rows = 4

for i in range(1, columns*rows):
    num = numpy.random.randint(bp.batch_size)
    image = bp.x_batch[num].astype(numpy.int)
    fig.add_subplot(rows, columns, i)
    pyplot.imshow(image)

pyplot.savefig(figure)
return figure
```



Training

```
import keras
from keras.models import Model, load_model
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator
from keras.applications.vgg16 import VGG16
import batchPreparation as bp

base_model = VGG16(weights='imagenet', include_top=False, input_shape=(bp.image_size, bp.image_size, 3

for layer in base_model.layers: layer.trainable = False

model = keras.models.Sequential()
model.add(base_model)

model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dense(1024, activation='relu'))
model.add(Dense(bp.num_class, activation='softmax'))
```

```python
print(model.summary())
```

```
Found 6899 images belonging to 8 classes.
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              25691136
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 8200
=================================================================
Total params: 41,463,624
Trainable params: 26,748,936
Non-trainable params: 14,714,688
_________________________________________________________________
None
```