

CPSC-8810 Deep Learning

TensorFlow for Image Classification Using VGG16

A project presented by:

Nishi Patel

Nandini Krupa Krishnamurthy

1. Description

The current trends in Image classification and object detection makes use of Deep Learning and Machine Learning methods. To learn about a large number of images, we need a model with large learning capacity. This project which is based on the same to classify a set of images using Convolution Neural Networks (CNN) conventionally for all tasks. A CNN is a supervised learning technique which needs both input data and target output data to be supplied. These are classified by using their labels in order to provide a learned model for future data analysis. CNN is one of the best methods to achieve our objective. CNN is composed of 3 subparts -Convolutional Layer, Pooling Layer and Dense Network. The dataset available project is a group of images from 10 categories - gossiping, isolation, laughing, pullinghair, punching, quarrel, slapping, stabbing, strangle and nonbullying. For this project, VGG16 is used for our project. VGG16 (also called OxfordNet), developed by the Visual Geometry Group from Oxford is a convolutional neural network architecture. It is considered to be one of the best vision Models. Hence, this project aims at classifying the given image into one of the 10 categories given using VGG16. In this process, our code accuracy is at 65%-70% on an average.

2. Network Structure

For this project, VGG16, a proper convolutional neural network model is used. This project is being implemented with 13 convolution layers with 13 batch normalizations with relu as activation function and 5 max pooling layers with size [2*2]. 3 fully connected layers along with optimizer - gradient descent is made use of here. To convert the last layer to a probability distribution softmax function is being used.

3. Training Strategy

a. Creating Data

The data is first divided into train and test data randomly in the ratio 8:2. The original had the 9 groups without nonbullying images. We have added them to be included in the dataset. All the images in the dataset are renamed accordingly with proper labels. The dataset looks similar to this:

```

\Dataset
  \train
    \gossiping
      gossiping_0.jpg
      gossiping_1.jpg
      .
      .
      .
    \isolation
    \laughing
    .
    .
    .
  \test
    \gossiping
      gossiping_4.jpg
      gossiping_7.jpg
      .
      .
      .
    \isolation
    \laughing
    .
    .
    .

```

We have created training and test data as images with fixed size (224*224). As the next step we have converted images into greyscale and stored as arrays of two types – images and label. In the last step, we have normalized the data

b. Training the data.

We have used tf.estimator for using estimators in the algorithm. VGG16 is used. We have 13 convolution layers. The code runs on the training data. Using session we are storing our model as .ckpt file after every epoch. We are training our model in batch size = 32 and number of epochs = 20. Here, we have made use of relu as activation function.

c. Test.py

We are taking image as command line argument and the reloading the saved model to predict the category of the image. The structure of test.py is same as that of train.py.

Note: The path for trained model has to be changed in this file in line 149.

4. Reference papers

[1] Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. "Tensorflow: A system for large-scale machine learning." In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265-283. 2016.

[2] See: <https://becominghuman.ai/paper-repro-learning-to-learn-by-gradient-descent-by-gradient-descent-6e504cc1c0de>

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

[4] Liu, Tianyi, Shuangfang Fang, Yuehui Zhao, Peng Wang, and Jun Zhang. "Implementation of training convolutional neural networks." arXiv preprint arXiv:1506.01195 (2015).

5. Code Usage Description

The packages are :

1. CV
2. OS
3. Numpy
4. Random
5. Tqdm
6. Shutil
7. Split-folders
8. Tensorflow

To train the data, use

```
python src/train.py
```

To test the model, use

```
Python src/test.py imagename.jpg
```