Image Classification Using Transfer Learning and Deep Learning

1 author:

Chitra G Desai
National Defence Academy
56 PUBLICATIONS **102** CITATIONS

Image Classification Using Transfer Learning and Deep Learning

Chitra Desai

Department of Computer Science, National Defence Academy, Pune

Abstract

Deep learning models have demonstrated improved efficacy in image classification since the ImageNet Large Scale Visual Recognition Challenge started since 2010. Classification of images has further augmented in the field of computer vision with the dawn of transfer learning. To train a model on huge dataset demands huge computational resources and add a lot of cost to learning. Transfer learning allows to reduce on cost of learning and also help avoid reinventing the wheel. There are several pretrained models like VGG16, VGG19, ResNet50, Inceptionv3, EfficientNet etc which are widely used. This paper demonstrates image classification using pretrained deep neural network model VGG16 which is trained on images from ImageNet dataset. After obtaining the convolutional base model, a new deep neural network model is built on top of it for image classification based on fully connected network. This classifier will use features extracted from the convolutional base model.

Keywords: Deep Learning, Convolutional Neural Network, Transfer Learning, VGG16, Image Classification

1. Introduction

Image classification problem in computer vision can be modelled conventionally, using machine learning, deep neural networks and in particular convolutional neural network [1], which are designed to work in isolation. If a model for classification is built from scratch by training it on a small data set, it is likely to get overfitted. Also, if the model is trained on large data set, it will consume huge compute resources, as well as, require days or weeks to train the model. However, this learning remains specialized to a particular domain and on a very specific dataset. As there is no knowledge retention in these conventional models, it cannot be transferred from one model to another. To overcome the problem of isolated learning paradigm, the concept of transfer learning [2] evolved.

Transfer learning allows to transfer the knowledge gained by previously learned task and apply it to similar another task [3]. With transfer learning the base network which consist of different layers depending upon the architecture is trained on the base dataset and the learned parameters are transferred to another network. Convolutional neural network models have layered architecture where different features are learnt at each layer. Therefore, transfer learning can be easily accomplished with convolutional neural network, where, the lower layer acts as a feature extractor and the final layers are used to extract more specific features.

This paper presents model for binary image classification on image dataset using pretrained model VGG16. The model implemented consist of two parts – convolutional base and classifier. The convolutional base will be used to extract the features. To extract features here, VGG16 is trained on ImageNet dataset [4] which contains over 1000 classes. Using these features the classifier will be trained to classify input image. The model for image classification is implemented using application module of Keras [5] trained on ImageNet dataset.

2. VGG16 Model

The Visual Geometry Group (VGG) from Oxford developed two convolutional neural network models one of 16 layers and the other of 19 layers [6]. They are widely known as VGG16 and VGG19. VGG16 accepts the input of size 224X224 pixels with 3 channels for RGB, though the input size is fixed here, it can take any value greater than 224. The input image is passed through convolutional layer, the filters here have 3X3 receptive field with stride 1. Pooling performed here is max-pooling with 2X2 filter and stride 2. The width of convolution layer is 64 in first layer and increases by factor of 2 till it reaches 512. All hidden layers use ReLu as its activation function. The stack of convolutional layer is followed by three fully connected layers. The final layer is the soft-max layer.

3. Dataset

The flower dataset consists of two types of flower images: daisy and dandelion. The dataset is extracted from [7]. The training dataset consist of 200 samples of each class and the test and validation dataset consist of 100 samples of each class. The training, validation and test data for daisy are stored in six different directories i.e., train data for daisy (200 samples), validation data for daisy (100 samples) test data for daisy (100 samples), train data for dandelion (200 samples), validation data for dandelion (100 samples) and test data for dandelion (100 samples). The image size of each sample is 224X224X3, here, 3 is for RGB channel. The sample image of daisy and dandelion is shown in figure 1.
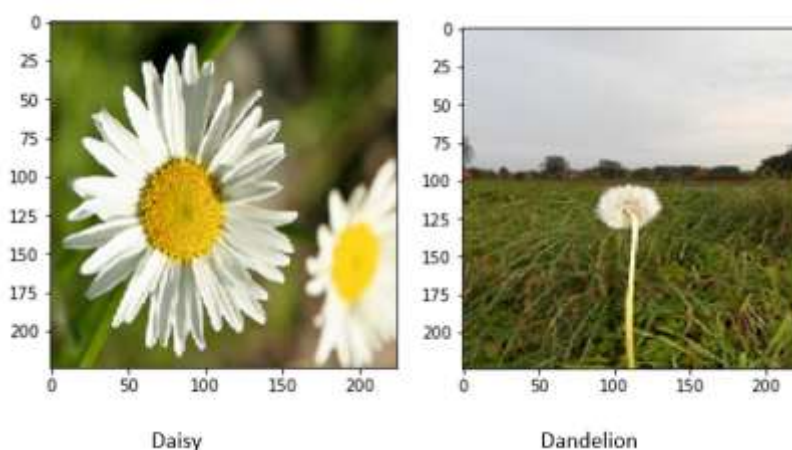

Figure 1 Flowers Dataset Sample (Daisy and Dandelion)

4. Pre-training and Feature Extraction

The convolutional base for convolutional neural network, VGG16 ConvNet is pretrained using ImageNet. The model summary for the pretrained model is as shown in figure 2. The fully connected output layers of the model are not loaded as it will not be used for prediction. Let us consider the name of this base model as *vgg16covmodel* (a user defined identifier). The shape of the output of *vgg16covmodel* is (7,7,512) as seen in the last block of figure 2.

Using this base model, the train, validation and test features and labels are extracted by applying random transformation to each image as it is passed to the model. In order to avoid the problem of memory overflow, the data is fed in batch size of 32 that is 32 images from the respective directories (train and test directories for daisy and dandelion) are fed to the model at a time. The shape of the extracted features and labels for train, validation and test is as shown in table 1.

```
Model: "vgg16"

Layer (type)                  Output Shape              Param #
=================================================================
input_8 (InputLayer)          [(None, 224, 224, 3)]     0

block1_conv1 (Conv2D)         (None, 224, 224, 64)      1792

block1_conv2 (Conv2D)         (None, 224, 224, 64)      36928

block1_pool (MaxPooling2D)    (None, 112, 112, 64)      0

block2_conv1 (Conv2D)         (None, 112, 112, 128)     73856

block2_conv2 (Conv2D)         (None, 112, 112, 128)     147584

block2_pool (MaxPooling2D)    (None, 56, 56, 128)       0

block3_conv1 (Conv2D)         (None, 56, 56, 256)       295168

block3_conv2 (Conv2D)         (None, 56, 56, 256)       590080

block3_conv3 (Conv2D)         (None, 56, 56, 256)       590080

block3_pool (MaxPooling2D)    (None, 28, 28, 256)       0

block4_conv1 (Conv2D)         (None, 28, 28, 512)       1180160

block4_conv2 (Conv2D)         (None, 28, 28, 512)       2359808

block4_conv3 (Conv2D)         (None, 28, 28, 512)       2359808

block4_pool (MaxPooling2D)    (None, 14, 14, 512)       0

block5_conv1 (Conv2D)         (None, 14, 14, 512)       2359808

block5_conv2 (Conv2D)         (None, 14, 14, 512)       2359808

block5_conv3 (Conv2D)         (None, 14, 14, 512)       2359808

block5_pool (MaxPooling2D)    (None, 7, 7, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
```

Figure 2 Model Summary for Pretrained VGG16 Model using ImageNet

Table 1 Shape of Train-Test Features and Labels

Features / Labels	Shape
train_features	(200, 7, 7, 512)
train_labels	(200,)
validation_features	(100, 7, 7, 512)
validation_labels	(100,)
test_features	(100, 7, 7, 512)
test_labels	(100,)

5. Model Building for image Classification

Once the train, validation and test features and labels are extracted using the pre-trained model another model is built for image classification based on fully connected network. This classifier will use features extracted from the convolutional base model.

The input shape to the sequential model is (7,7,512). The activation function at hidden layer used is ReLu, dropout is set to 0.5. Dropout [8] is used to address the problem of overfitting. The output layer activation is set to sigmoid. The model summary is as shown in figure 3. The model is compiled using Adam optimizer.

As the problem is of binary image classification, the loss is computed using binary cross entropy. The model is fitted using the train features and labels, validation data with batch size 32 and 100 epochs. The model accuracy is as shown in figure 4

```
Model: "sequential_12"

Layer (type)                Output Shape              Param #
=================================================================
flatten_12 (Flatten)        (None, 25088)             0

dense_32 (Dense)            (None, 256)               6422784

dropout_2 (Dropout)        (None, 256)               0

dense_33 (Dense)           (None, 1)                 257
=================================================================
Total params: 6,423,041
Trainable params: 6,423,041
Non-trainable params: 0
```

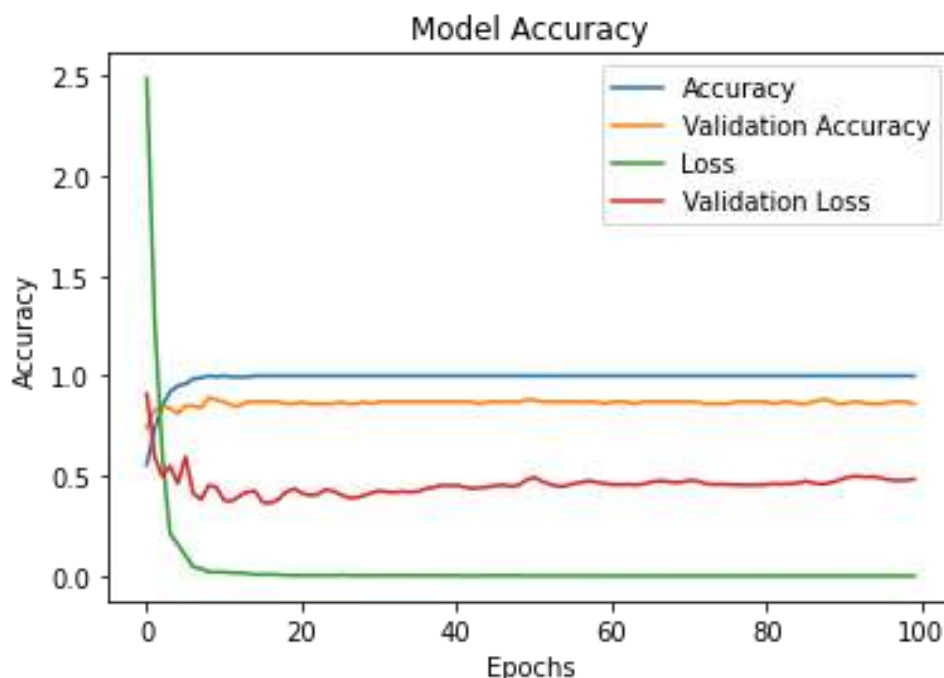Figure 3 Sequential Model (DL) for Classification

Figure 4 Model Accuracy and Loss

6. Experimental Results

The model on fitting was used for making predictions on test data. The predictions were made for 10 random samples from test directory at each execution. The experiment was repeated five times and the following results were obtained.

Table 2 Classification Result

Execution	Out of 10 Random Samples at each Execution	
	Rightly Predicted	Wrongly Predicted
1	8	2
2	8	2

3	10	0
4	9	1
5	10	0

Thus, it can be stated that the model has achieved 90% of accuracy for the flower data set image classification.

7. Conclusion

Convolutional Neural network are widely used in image classification. With the pre trained models, by applying transfer learning it has become possible to train the model with limited resources. In fact, the pretrained models avoid reinventing the wheel and also avoid building the model from scratch. It indeed helps focus on the capturing more specific features by adding more layers specific to the task. In this paper, the model is trained using VGG16 to extract features and a sequential model is built on top of it to perform classification of images. The data set used is a balanced data set with equal number of samples in each class and hence no bias. The results show 90% of accuracy of model when executed on 10 random samples with five iterations using CPU system, batch size of 32 and 100 epochs. The experiment was repeated by replacing VGG16 model with VGG19 on the same dataset. It was observed that there was no remarkable change noted in accuracy for classification.

References

1. Y. LeCun et al., Backpropagation Applied to Handwritten Zip Code Recognition, in Neural Computation, vol. 1, no. 4, pp. 541-551, doi: 10.1162/neco.1989.1.4.541, Dec. 1989
2. Caruana, R.; Silver, D. L.; Baxter, J.; Mitchell, T. M.; Pratt, L. Y. & Thrun, S. , 'Learning to Learn: Knowledge Consolidation and Transfer in Inductive Systems' ,1995
3. Karl Weiss* , Taghi M. Khoshgoftaar and DingDing Wang, A survey of transfer learning, Journal of Big Data, Weiss et al. J Big Data , 3:9, DOI 10.1186/s40537-016-0043-6, 2016
4. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale HierarchicalImage Database. InCVPR09, 2009.
5. Chollet, F., keras, 2015, GitHub. https://github.com/fchollet/keras
6. K Simonyan, A Zisserman , Very deep convolutional networks for large-scale image recognition - arXiv preprint arXiv:1409.1556, 2014
7. TensorFlow Team, Introduction to TensorFlow Dataset, Feb 2019 https://www.tensorflow.org/tutorials/images/classification#download_and_explore_the_dataset
8. Hinton, Geoffrey E., Nitish Srivastava, A. Krizhevsky, Ilya Sutskever and R. Salakhutdinov. "Improving neural networks by preventing co-adaptation of feature detectors." ArXiv abs/1207.0580 , 2012