Final Project Report: Image classification task

Pranay Barkataki
pranay.barkataki@gmailcom

Ashish Kore
ashishkore93@gmail.com

In this report, we briefly discuss the details of various models used for the task of image classification for the given set of data. We have extensively discussed the hurdles which we will tentatively face during the image classification, difficulties ranging multiple objects in an image to limitation of free computation servies. Finally, we give a detailed explaination on the possible aprroaches and final models which are selcted based on the performance of model with respect to accuracy and loss of both train and validation data sets. We have reached upto 62.28% accuracy in test set, by using deep Convolutional Neural Networks which we have briefly discussed in this report.

1. Introduction

Deep learning is growing field interms of its capability in various fields, one of them is image classification. Image classification is the task of classifying a group of pixels or matrices (2D or 3D) in their respective category. The task of image classification are broadly divided into supervised and unsupervised classification. However, the image classification project in which we are working on falls into the catergory of supervised classification.

The task of image classification is a supervised learning algorithm, where there is set of distinct target classes, and we have to train our neural network model to classify the images into one of these classes. Specifically, convolutional neural network (**CNN**) [1] revolutionized the field of image classification. The CNN takes the input as an image, and then assigns importance to numerous objects in the image by learning weights and biases, which ultimately leads to classification of the images.

The project of '**Image classification task**' has 200 distinct classes and in the training dataset there arewith 90,000 images with 450 training images for each class. There are 50 images for each class for both validation as well as test dataset. Each image has resolution 64x64, which may be difficult to classify some images for the human eye. In following section we dive deep into possible hurdles we may face during the classification task.

2. Difficulties faced during image classification

Figure 1. Different type of objects in a single class of images

On the bird's eye view of the images in the training dataset, we notice that in each class there are some images which has both target object as well as other non-target objects. For example in the training class **n01742172** which 450 images of the snake *boa constrictor*, in

the image **n01742172_2.JPEG** there is a girl (non-target object) holding a boa constrictor (target object) also in image **n01742172_1.JPEG** there are group of three boys holding boa constrictoras as shown in figure 1. Though in the description of the project, localization of objects in an image is not to be done. However, with the presence of multiple objects in an image, it makes the classification task quite hard. The difficulty of image classification becomes more intricate with the variations of *size*, *color* and *orientation* of the target object in the images. The resolution of the images is just **64x64** pixels, which will make feature extraction a difficult task. Even it is difficult for a human eye to completely detect multiple objects in an image. Due to the complexity of the problem, training any CNN model would require lot off computational time, and the biggest hurdle is that low computational free services such as google colab, and kaggle offers only 12 hours per day and 30 hours per week computation time, respectively. This leads us to a very important question, what are the possible approaches to be taken to solve this image classification problem ?

3. Transfer learning method

Before venturing into the different models based on transfer learning method [2], we briefly discuss the transfer learning method. In the year 1993, Lorien Pratt wrote the first paper [3] on the transfer learning method. It is a machine learning method, where a model developed for a task is reused as the starting point for a model for similar other tasks. The pretrained models which we have used as the base model, for the purpose of image classifications are listed below,

1) VGG16 [4].
2) VGG19.
3) ResNet50 [5].
4) ResNet101.
5) ResNet152.
6) ResNet50V2.
7) ResNet101V2.
8) ResNet152V2.
9) DenseNet121 [6].
10) DenseNet169.
11) DenseNet201.

All the above models are the pretrained ImageNet models. ImageNet is a large labeled dataset of real-world images. It is one of the most widely used dataset in latest computer vision research. We removed the last output layer (softmax) from all the above pretrained models because we have only 200 distinct classes of images, which is much less as compared to the number of distinct classes on which ImageNet models are trained. After some extensive literature review, we understood that to fine tune the pretrained models for our dataset we have to add a layer global average pooling, and then add two fully connected layers, which is followed by a dropout layer connected to a softmax classification layer of dimension 200. We have mainly chosen three optimizer for our customs models,

- RMSProp [7].
- SGD [8].
- Adam [9].

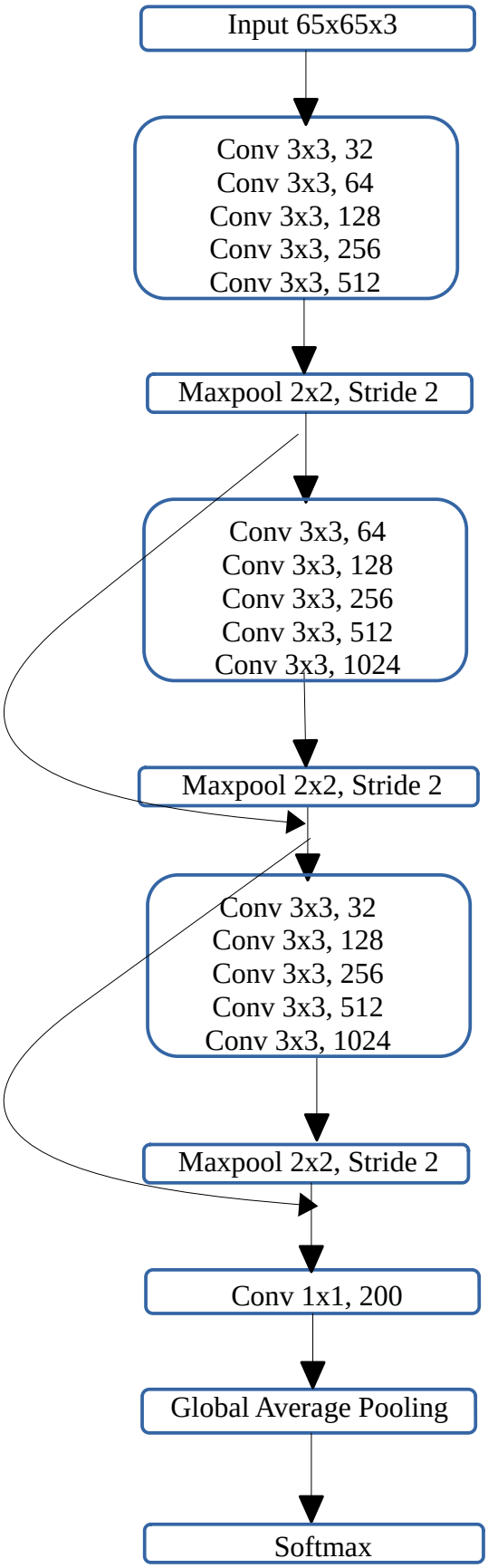

Figure 1: Network 1 architecture.

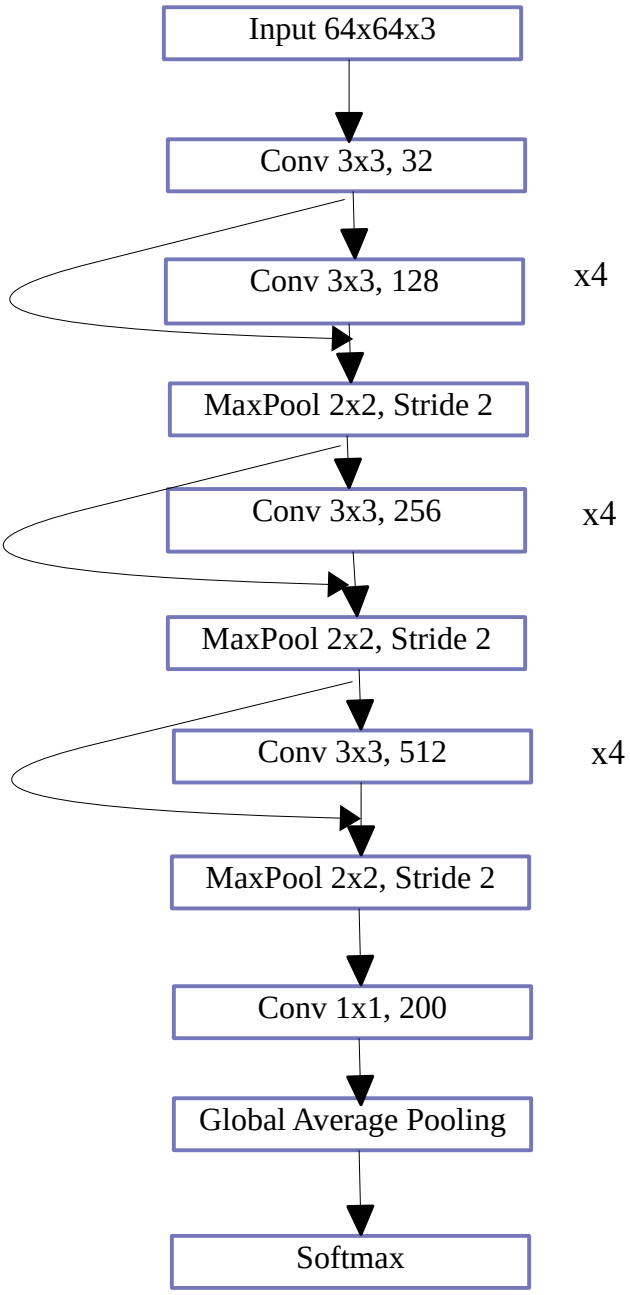

Figure 2: Network 2 architecture

We tried numerous different permutation with respect to the dimension of the two fully connected layers, different activation function for the dense layers, different value of dropout probability, and tried all the three specified optimizers. We have all this permutations for the 11 pretrained models which we have discussed before. Out of all these permutations, the model which gave the best result was the ResNet101 model. It was the base model on which we have added global average pooling layers, followed by two dense layer of dimensions 1024 and 512, and the activation function for both of the dense layer applied is ReLu. On top of these dense layers we added dropout layer with probability 0.4, and finally we added the softmax output layer. The test set accuracy for this customized model was around 35 %. In order to increase the test set accuracy, we did a more extensive literature survey, and came across the following reference [10], where the authors discusses two custom build CNN model for image classification. The architecture for both of the CNN models are shown in Figure (1) and Figure (2). In the ref. [10], the models classifies into 200 distinct classes, and in the training dataset there are 500 images for each class. Number of classes and size of the dataset is almost similar to the task in our hand, therefore both of the CNN can be potential approaches for our image classification task.

4. Work flow of DenseNet networks:

For the begining we started with transfer learning approach based on ResNet and VGG as they perform better for computer vision task. The maximum validation accuracy we achieved using ResNet101 is 35%. After achieving this maximum accuracy model gets saturate without further increase in accuracy which is due to depthness of these transfer learninng models, and with very deep neural networks there is a problem of vanishing gradients. This problem is rectified by using residual networks, which are used as a shortcut connection to pass input from one block to another. This concept is implemented in Network 1 and Network 2, as shown in Fig.(1) and Fig.(2). The architecture of the network 1 and network 2 is dicussed are discussed in detail in the following two sections.

4.1. Network 1:

The architecture of the network is discussed in detail below,

i. We built a custom architecture of 3 blocks having 5 convolution layers of varying channels and 1 MaxPooling layer in each block.

ii. Just before applying concatenation we applied space_to_depth to make spatial dimensions of both the layers are equal before concatenation.

iii. We concatenate the skip connections from the output of each block with the output of the next block, preserving the information from both the blocks before being fed to the subsequent block.

iv. The final layer in the model have 1x1 convolution layer and to averages the spatial dimensions of a matrix of any size we used a GlobalAveragePooling layer. This layer gives us the ability to design a model which can take input image of any size.

Batch normalization is applied to every convolutional layer, as seen in Fig.(1). The batch normalization helps us to normalize the inputs of the preceeding layer. This procedure ensures that the activation functions (ReLU) of our models do not get skewed at any particular point, and also increases the speed of computation. The process of training the network1 model is listed below,

1) We first train the model in low resolution images 32 x 32, for 20 epochs. The validation accuracy saturates around 28 %.

2) We then train the model in 64 x 64 resolution images, for 20 epochs. The validation accuracy staturates around 44%.
3) We apply image augmentation horizontal_flip=True, width_shift_range=0.2, height_shift_range=0.2. After training the model for 90 epochs the model staturates at validation accuracy of around 62%.

We have applied Adam optimizer with a default learning rate of 1e-3, and it showed very good results. On the submission of the test set results, this model finally gave the test set accuracy 61.95%. In the next section we delve deep into understanding the network 2.

4.2. Network 2:

Description of netwwork 2 is as follows:

i. We replaced the first convolution layer that initially consisted of 64 (7x7) filters with stride (2,2), by 32 (3x3) filters with stride (1,1) and removed the max pooling layer from ResNet50.

ii. We removed the 1st block consisting 4 convolution layers of 64 (3x3) filters. Also removed the skip connections after every 2 convolution layers instead maintained it between every 4 convolution layers which is one block here. We replaced the original add function in shortcuts with concatenation so that it preserves the channels from the previous block and not merge them. We added a Batch Normalization and ReLU activation layer after each shortcut.

iv. As per requirement of the project, we replaced final Fully connected layer with 1x1 convolution layer for decreasing the number of channels to the required number of classes followed by a GlobalAveragePooling layer.

Here the one block consisting of 4 convolution layers, for each block we kept the channel size remains same for all 4 layers in the corresponding block. First block of model contains only one layer which includes padding which we set as ‘same’ to get output of same dimension as that of the input dimension. After that Batch normalization is applied subsequently which is followed by ‘ReLu’ activation for speed up the task without affecting the performance of network. Training process for network 2 is as follows:

1) For training the network we have used ‘RMSprop’ optimizer for faster convergence with smaller learning rate of 0.0001 with epsilon 1e-08. Also have kept fixed steps_per_epochs as 512 throughout.

2) We kept the input size fixed i.e. 64x64 through out the run, we run the whole network in steps of 30 epochs in one run due to computational limitation on google colab. So for first 30 epochs the validation accuracy of network 2 is reached maximum to 38 % for which weights are saved and loaded for next set of epochs.

3) In second step of epochs validation accuracy get rises nearly to 45 %, in third step 49 %, in fourth to 52 %, in fifth to 56 %, in sixth to 59 % and in seventh 62 %.

4) Finally once we observe that validation accuracy got saturated without further improvement we used ‘adam’ optimizer and followed by ‘RMSprop’ optimizer, and finally we got improved validation accuracy of 63.47% .

After implementing this network on test set we got accuracy of 62.28% which is highest as compared to network 1.

6. Summary and Conclusion

In this project, we have analysed the provided data in extensive detail. Based on our analysis we have extensively discussed the difficulties involved in image classification task, and

provided suitable examples for the better understanding of the difficulties, wherever necessary. We then briefly discussed the transfer learning method, and based on this method what are the possible approaches we have choosen in regard to the task in hand. We then extensively discussed the CNN network 1 and network 2. We then extensively discuss the training procedure of both the networks, and finally we are to conclude that the network 1 reported highest value of the validation accuracy, and final value of the test set accuracy for network 1 is 62.28%.

References

[1] LeCun, Yann, et al. “Backpropagation applied to handwritten zip code recognition.” Neural computation 1.4 (1989): 541-551.

[2] West, Jeremy, Dan Ventura, and Sean Warnick. “Spring research presentation: A theoretical foundation for inductive transfer.” Brigham Young University, College of Physical and Mathematical Sciences 1.08 (2007).

[3] Pratt, Lorien Y. “Discriminability-based transfer between neural networks.” Advances in neural information processing systems. 1993.

[4] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[5] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[6] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[7] Tieleman, Tijmen, and Geoffrey Hinton. "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning." University of Toronto, Technical Report (2012).

[8] Ning Qian. On the momentum term in gradient descent learning algorithms. Neural networks : the official journal of the International Neural Network Society, 12(1):145–151, 1999.

[9] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[10] Abai, Zoheb, and Nishad Rajmalwar. "DenseNet Models for Tiny ImageNet Classification." arXiv preprint arXiv:1904.10429 (2019).