Scene Classification with Deep Neural Nets using Background Suppression

Manoj K,
Research Associate,
PES University,
Bangalore, India
e-mail: mnjkshrm@gmail.com

Shylaja S S,
Chairperson, Dept. of CSE
PES University
Bangalore, India
e-mail: shylaja.sharath@pes.edu

***Abstract*— Deep Learning can be a powerful tool to replace the human eyes in the field of Scene classification. In this paper, we propose scene classification for four different classes using background suppression and Convolutional Neural Network. Background suppression is achieved with luma transforms. Supervised learning is used and the processed images are fed to the four layered convolutional neural network, which would enable the system to classify the images.**

Background suppression seems to indicate an increase in the specificity which is very much essential for scene classification. However, with increase in layers, sensitivity of the system does increase resulting in lesser validation loss. The training and validation accuracies have shown much improvement in comparison with the convolutional neural network and several other approaches.

Keywords- Deep Learning, Scene Classification, Applications of Computer Vision, Image Processing, Convolutional Neural Networks

I. INTRODUCTION

Scene classification is one of the dominant domains of problem solving in the field of Computer Vision. A collection of various objects and features, which combined with background noise make up a 'Scene'. The human brain is well-trained in distinguishing various kinds of scene. It can process the image and classify it, regardless of the background noise as to whether the person is looking at a building or a beach or a forest.

Scene Classification in general is recognizing the various attributes or features of a particular scene and providing information. Scene classification can be approached in multiple ways, by object recognition and classification or by using feature extraction or feature matching. Scene classification is an excellent application in various sectors. It can be used in agriculture, to detect the growth of crops or detect diseased crops. It is used in military, to recognize intruder aircrafts or suspicious activities. Also it serves in manufacturing industries for quality inspection.

Deep Learning is a subset of Machine Learning, with layers of neural networks (Multilayer perceptron). In recent developments, they have proven to be more efficient than biological neurons. Deep Neural Nets are layers with inputs, output(s) and several hidden layers in between the input and output layer. Once the network is trained, the system is very efficient to perform any classification/regression. The Convolutional Neural Network (CNN) is a neural network with learnable weights and biases. Using a ConvNet solves more problems faced by a regular neural network. In a convolutional neural network, there is an Input Layer, a convolutional layer, a pooling layer, an activation layer, and a fully connected layer. Pre-processing of images is necessary before they are being fed to the input of the network. Pre-processing involves changing the dimensions of the image, histogram equalization and other techniques. Then the features of an image, like pixel level features such as colors or location, local features such as edge detection or image segmentation and Global features such as the entire image or a template of the image are extracted.

This research paper showcases how the use of feature extraction techniques on pixel level features, in order to suppress the background noise, can enhance the efficiency of Deep Neural Nets. The main objective is to achieve maximum specificity of an image by eliminating background noise thereby isolating it from the other classes.

II. PREVIOUS WORK

'Scene Classification in Images' (Dutt, B V V Sri Raj et al. 2009) involves separating characteristic vector features, by taking into account the diagnostic information stored in power spectrum of each category of image and then using Supervised Learning to train the system. The paper emphasizes more on the structural discrimination vectors between the scenes of an image and the pre-processing method involves reducing illumination effects. Which is contrast to the research work shown in this paper, where in the image goes through a change in color space, hence bringing out the most dominant color in the image. With reduction in the image illumination, we decrease the pixel level features which help in finding the image. Another paper on 'Scene Classification with Low-dimensional Semantic Spaces and Weak Supervision' (Rasiwasia, Nikhil and Nuno Vasconcelos. 2008 IEEE Conference on Computer Vision and Pattern Recognition (2008): 1-6) reveals about weak supervised training where, images are represented as vectors of posterior theme probabilities. Each image theme has a probability density on the space of low-level features. In this approach images are associated with multiple themes, even though they are not associated to multiple labels. 'Object-Scene Convolutional Neural Networks for Event Recognition in Images' (arXiv:1505.00296) proposes a different architecture where two deep nets are used for object and scene classification. Here the ImageNet dataset is used for training the objects network and Places Dataset is used

for Scene networks for extracting the information from the scenes, which is trained using the AlexNet architecture.

III. DATASET

Dataset generation for a particular class of images is not simple. So for a custom class of images, getting the right dataset is very important. For this research work, four different scene categories are selected, i.e. Aero planes, Buildings, Factories and Farm Fields. Along with the existing dataset, Image augmentation was done for a few set of image classes like the factories, fields and buildings, in order to enhance the dataset.

Image augmentation has been performed by changing color space, Histogram equalization which results in images with better clarity with even and uniform distribution of color intensities. Another way of generating large number of images involves rotating at various angles, variation in brightness and saturation.

The Scikit-learn also provides a wide range of libraries which help in augmenting the image in order to generate more data. As seen in Figure-3.2, the image is rotated by a certain angle, which is repeated for other images, thus generating a huge dataset. The end result is obtaining a dataset of 800-1000 images for each class. Among these dataset, 80% images are used for training and 20% are used as a validation set.

Figure-3.1

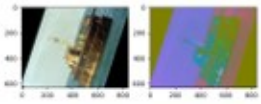

Figure-3.2

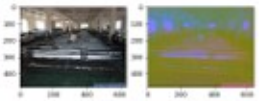

Figure-3.3

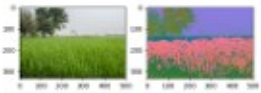

Figure-3.4

A histogram is a graphical representation of the pixel values of red, green and blue. This is done by converting the image to HSV color space and equalizing the histogram of the other channels and again converting the image to BGR color space

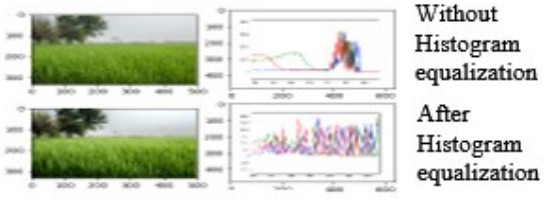

Figure-3.5

IV. PROPOSED METHOD

For this particular research work, we have considered enhancing the output or results against a regular neural network, with background suppression. By eliminating the unwanted background noise, we can achieve better accuracies.

We have already pre-processed the image with histogram equalization. Now we have to eliminate the background. The most prominent part of the image would be retained and the rest is suppressed. So by extracting the maximum dominant color from this input image, we can get the major or the most prominent part of the image.

As seen in Figure-3.5 green is seen to be the most dominant color. So we begin by reducing the number of colors in the image using the Image Palette, we make use of Adaptive Image mode for quantizing the colors and 8-bit pixel color mapping. The maximum number of dominant colors allowed is only one; maximum allowed value would be 256. We then add an alpha mask(0) to the image, to change it to RGBA format. Now with this, we have to make use of the luma transform which converts the image to its YUV color space. The U & V values are mapped to appropriate BGR values using custom Look up table (color map) generated as seen in figure 4.1, which is nothing but a progression of the color from green to blue.

$$Y' = 0.299R + 0.587G + 0.114B$$
$$U = 0.492(B - Y')$$
$$V = 0.877(R - Y')$$

Figure-4.1

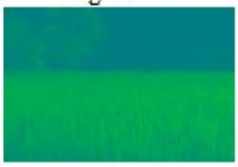

Figure-4.2

We iterate through the entire image (*width* x *height)* thereby returning a tuple of repetitions and the maximum color across the entire image. This will now result in getting the maximum dominant color in the image.

Figure-4.3

Now since we have our dominant color, we can save this is an image and proceed ahead for the background suppression.

From the original image, we split the channels of the image to R, G and B. Since the color distribution in Figure-4.3 is even throughout the image, we take the first index and split the values of R, G and B here also. Now we can set a threshold of certain value above and below the value of color channels. For this work, two threshold values were tested.

Blue [value] ± 10 and Blue [value] ± 20
Green [value] ± 10 and Blue [value] ± 20
Red [value] ± 10 and Red [value] ± 20

The value of each channel ranges from 0-255. From this experiment, the threshold value was decided to be 20 above and below the channel color. After this process is completed, we achieve a background suppressed image.

Figure - 4.4

As seen in Figure-4.4 most of the background is suppressed for the image. This can also be seen in other classes too.

Figure-4.5

Figure-4.6

Figure-4.7

The pre-processed images are trained in a batch of 26 images. The images are categorized and labeled with their classes respectively.

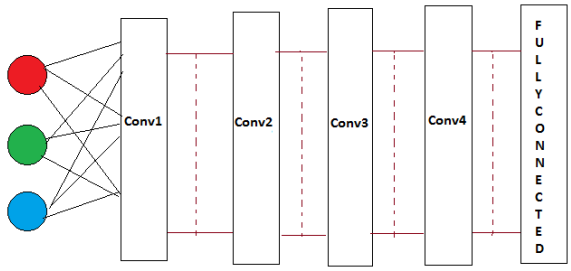

Figure-4.8

Layers	*No. of Filters*
Input	3 channels
Conv1	16
Max Pool	
Conv2	32
Max Pool	
Conv3	32
Max Pool	
Conv4	64
Max Pool	
Fully Connected Layer	128

Table – 4.1

As we can see in Figure 4.9, the neural network implemented has lesser number of Convolutional Layers with 3 inputs, Red, Green & Blue channels and also one Fully Connected Layer.

Since the network is not too deep, we do not face the issue of overfitting, hence a dropout layer is not necessary.

The Tensorflow Network is fed with input images of size 128x128x3, which are split into 3 channels respectively R, G and B. Each of these channels is input to the next convolutional layer, which has 16 filters of kernel size 3x3. The filters are the weight matrices, which are initially seeded arbitrarily, and are initialized with *tf.truncated_normal* with a standard deviation of 0.05 from the mean. These weights are then corrected during the training through Back-propagation. At the end of output layer, ReLU activation is used for introducing non-linearity into the network.

$$f(x) = \max(x, 0) \quad , where\ x\ is\ input$$

The output is again fed to the next hidden/convolutional layer (Conv2) which has 32 layers with a kernel size of 3x3. The output of this is again fed to next Convolutional layer (Conv3), which is fed to next convolutional layer with 64 filters (Conv4). The output of each convolutional layer is maxpooled which plays a major role in down-sampling. Since Max pooling results in extreme features, the resulting image could be of much rich features rather than average pooling which takes into account all the values. The output of final Convolutional layer is sent to a flattening layer which reshapes the tensor. The flattened tensor is now fed to the Fully Connected Layer for the classification of the features extracted by the convolutional layer. Then we add a Softmax Layer which helps in determining the probability of each class.

For this research two optimizers were chosen among which the Adam Optimizer resulted in much better accuracy and less validation loss.

Adam Optimizer with a learning rate of 1e-04 is used during the tf.session, by which weights are re-adjusted during training. The RMS Optimizer resulted in a Training Accuracy of 84.5% and Validation Accuracy of 78.3%.

The Adam Optimizer resulted in a Training Accuracy of 100% and Validation Accuracy of 96.2% for 162 epochs.

V. RESULTS

The Convolutional Neural Network was tested for this particular Multiclass Classifier for a test data set for each class. The results for both the networks with and without the proposed method are tested and the values for Specificity and Sensitivity are verified against the same test data set. The Network which is trained with a background suppressed image achieves better results than a regular Neural Network where input data is not pre-processed.

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$Specificity = \frac{TrueNegative}{TrueNegative + FalsePositive}$$

Without Background Suppression – Table 5.1

	Aero plane	Buildings	Factories	Fields
Sensitivity	*50.5%*	*41.57%*	*87%*	*92.75%*
Specificity	*98%*	*98.35%*	*73.12%*	*95.77%*

With Background Suppression – Table 5.2

	Aero plane	Buildings	Factories	Fields
Sensitivity	*71.5%*	*82.865%*	*75%*	*85.8%*
Specificity	*99.49%*	*92.76%*	*96.34%*	*99.56%*

With proper inspection of both Table-5.1 and Table-5.2, we can conclude that the neural network model has a very good Specificity when the background is suppressed than the neural network model which has been trained without background suppression.
For calculating the values of True Positive, True Negative and False Positive, False Negative a threshold of 60% of the right prediction is considered.

VI. Discussions

This research work showcases an important aspect of Deep Learning as already discussed, i.e. Deep Neural Networks are only as efficient as the one that trains it and Deep Neural Nets require a large number of training dataset to achieve good accuracy. Hence with lesser dataset, the probability of getting the right prediction for an image is less.

In comparison with the work on Spatial Pyramidal Matching for Recognizing Natural Scenes [10], the method of Background Suppression eliminates the necessity for SIFT Descriptors in the Feature Extraction method. The color quantization method in this paper is better than making use of k-means clustering on random patches which is applied on the training data [10].

The method used in Scene Classification with Convolutional Neural Networks [16] proposes the use of ResNet 34 architecture, which is a better alternative to the Baseline architecture. But the results of Baseline architecture, can be improved when we suppress the noise in the background.

With pre-processing the images before they are fed into the neural network, the isolation of the respective class of the new image among its other classes becomes very much simpler. This method can be applied to various fields for quick prediction without false results. It can be of astute use in manufacturing industries, where the defects in manufactured items can be easily identified. The worst case scenario might occur for scenes with dataset which includes human figures, which would result in additional noise being added and many a times not separable.

Increase in the layers also results in increase in the accuracy of the system. But as already seen, with background suppression, the idea of multiple hidden layers can be discarded. If the dataset is increased, we might also have to vary the training-batch size of the images. Memory of the system is to be considered during training if dataset increases. The size of the image is also a memory constraint. Hence the size of images for training & validation is 128x128. With the training carried out on a GPU, the time taken for 162 epochs reduces drastically.

The Adam Optimizer has the lowest training loss compared to the RMS Optimizer. The gradient descent optimizer results in underfitting and has poor training accuracy. So using an Adam Optimizer for images with a learning rate of 1e-04 is considered to be a good practice. A hybrid model with the Gradient Descent and Adam Optimizer can be used for the further research of this work.

References

[1] Dutt, B V V Sri Raj, "Scene Classification in Images", et al. 2009

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database", Dept. of Computer Science, Princeton University, USA.

[3] Cheng, G., Han, J., & Lu, X., "Remote Sensing Image Scene Classification: Benchmark and State of the Art", Proceedings of the IEEE, 105, 1865-1883.

[4] Bandhu, A, & Roy, S.S. (2017), "Classifying multi-category images using deep learning: A convolutional neural network model", 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 915-919.

[5] Perez, L., & Wang, J. (2017), "The Effectiveness of Data Augmentation in Image Classification using Deep Learning", CoRR, abs/1712.04621.

[6] Fidler, S., Urtasun, R., & Yao, J. (2012), "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation", 2012 IEEE Conference on Computer Vision and Pattern Recognition, 702-709.

[7] YIN Xiangnan, CHEN Weihai, "Fine-tuning and visualization of Convolutional Neural Networks", Beihang University, Beijing 100191.

[8] Dirk B. Walther, Eamon Caddigan, Li Fei-Fei and Diane M. Beck, "Natural Scene Categories Revealed in Distributed Patterns of Activity in the Human Brain", Journal of Neuroscience 26 August 2009, 29 (34) 10573-10581.

[9] Fredrik Lundh & Matthew Ellis, "Python Imaging Library Overview", March 12, 2002, www.pythonware.com/media/data/pil-handbook.pdf

[10] Spatial Pyramid Matching for Recognizing Natural Scene Categories:https://hal.inria.fr/file/index/docid/548585/filename/cvpr06_lana.pdf

[11] Scene classification with Convolutional Neural Networks: http://cs231n.stanford.edu/reports/2017/pdfs/102.pdf