International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Image Classification using Supervised Convolutional Neural Network

Saripalli Sri Sravya, Kalva Sri Rama Krishna, Pallikonda Sarah Suhasini

Abstract*: Deep learning algorithms, in particular Convolutional Neural Networks have made notable accomplishments in many large-scale image classification tasks in the past decade. In this paper, image classification is performed using Supervised Convolutional Neural Network (SCNN). In supervised learning model, algorithm learns on a labeled dataset. SCNN architecture is built with 15 layers viz, input layer, 9 middle layers and 5 final layers. Two datasets of different sizes are tested on SCNN framework on single CPU. With CIFAR10 dataset of 60000 images the network yielded an accuracy of 73% taking high processing time, while for 3000 images taken from MIO-TCD dataset resulted 96% accuracy with less computational time.

***Index Terms*: Deep learning, supervised convolution neural network (SCNN), image classification, supervised learning.**

I. INTRODUCTION

Classification of images has been a vast research area in computer vision as well as machine learning that aims at classifying objects existing in images into significant classifications. Deep learning has been a popular research issue in past few years, and convolutional neural network is the common approach for image recognition, image classification, picture clustering. As a state-of - the-art consequence, CNN is been created for both image recognition as well as classification.

The aim of the object categorization is to define the labels for the images provided and given a new image it should recognize to which group it belong to. There are several recognition techniques such as K-Mean, Nearest Neighbor Classifier, K-Nearest Neighbor (K-NN), Convolutional Neural Network (CNN), Support Vector Machine (SVM), etc. Image classification is an effective research field in which common implementations such as self-driving vehicles and automatic robots have been researched.

II. RELATED WORK

Initially, in order to perform image classification texture [1] had been one of the important characteristic. Textural features can be computed for classification. SVM (Support Vector Machine) classifier played vital role in image classification based on different applications. When it comes to multi-category image classification LP (Linear Programming) method based on Mangasarian approach combined with QP (Quadratic Programming) method based on SVM approach is used [2]. With its high generalization performance, the support vector machine (SVM) method is perceived as a good candidate without adding a priori knowledge, even though the input space dimension is quite high i.e. SVM could generalize on problematic image classification problems with images having high dimensional histogram features [3].

In the case of remote sensing data, SVM is chosen as a classifier [6]. When the classification has to be performed on the images having low-level visual features has been a problem in content-based image retrieval in order to solve these Bayesian classifiers has been used [4].

Deep learning has become one of the prominent approaches for image classification. K-nearest neighbors approach had been used for image classification when features that had to be extracted from images are color, texture and regions [5]. Kernel based approaches like Reg-RBFNN (regularized radial basis function neural networks) and Reg-AB (regularized AdaBoost) had been chosen for hyper spectral image classification [7].

PCA network (PCANet) i.e. is cascaded principal component analysis network which is a prefixed, highly hand-crafted, carefully learned network. PCANet turned into an efficiently designed and easy approach for image classification that has been tested on different benchmark datasets [8]. Recurrent Neural Networks (RNNs) along with Convolutional Neural Networks (CNNs) was used for addressing the label dependencies present in typical multi-label image classification [9].

III. SCNN FRAMEWORK

Convolutional Neural Networks (CNN), suggested by Yann LeCun in 1988, is a unique design of artificial neural networks which utilizes certain visual cortex characteristics. Image classification is the popular application of CNN. In a supervised learning model, a labeled dataset is used, so that it can be used by the algorithm to evaluate its accuracy on training data.

Convolutional layer: A CNN's key construction block is none other than a convolutional layer. The hyper parameters of the layer consist of a collection of learnable filters (or kernels) with a tiny receptive field but spread out through the input volume's complete depth. Each filter is slided over the width and height of the input volume during the forward pass, calculating the dot product among the filter entries and the input,

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Saripalli Sri Sravya*, Student, Department of ECE, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India
Kalva Sri Rama Krishna, Professor, Department of ECE, Velagapudi Ramakrishna Siddhartha Engineering College , Vijayawada, India
Pallikonda Sarah Suhasini, Associate Professor, Department of ECE, Velagapudi Ramakrishna Siddhartha Engineering College , Vijayawada, India.

resulting the creation of a 2-dimensional activation map of that filter. Assembling filters the activation maps formed along the depth dimension is the convolution layer's full output volume.

The dimensions of the output volume of the convolution layer are controlled by 3 hyper parameters: depth, step and zero-padding. The output volume depth regulates the total number of neurons present in a layer concerning to the same input volume area. Stride is one of the hyper parameters which regulates the allocation of depth columns (width and height) around the spatial dimensions. When the step is 1 we migrate one pixel at a moment of the filters which results in highly overlapping receptive areas among the columns, as well as large volumes of output. .

In the same way, the filter is rendered by S units at a moment per output for any integer S >0 a stride of S. where S >= 3 stride lengths are unusual. Occasionally padding the input with zeros at the input volume boundary is useful. The third hyper parameters is the size of the padding. With this padding, spatial size control of the output quantity can be obtained. It is possible to calculate the spatial dimensions of the output volume as a function of the input volume size W, with K as the kernel field dimension of the neurons present in the convolutional layer, with S being the stride, and the last zero padding P added on the border. In order to determine the number of neurons that "fit" in a specified volume is calculated by using

$$\left(W - K + 2P/S\right) + 1 \qquad (1)$$

$$P = (K - 1)/2 \qquad (2)$$

ReLU layer: ReLU is the acronym of the rectified linear unit, where non-saturating activation function is being applied. By setting them to zero, it efficiently deletes adverse values from an activation map. It improves the nonlinear characteristics of the decision function and the network as a whole without influencing the convolution layer's receptive fields. Here saturating hyperbolic tangent and the sigmoid function are also used to boost nonlinearity.

$$f(x) = max(0, x) \qquad (3)$$

Pooling layer: One more imperative conception of CNNs is pooling, a non-linear down-sampling type. The reason for max pooling concept in convolutional neural networks is to gradually decrease of representation's spatial size, decrease the of parameters count, memory footprint and quantity of calculation in the network, and consequently regulating overfitting. Inserting a pooling layer once in a while between consecutive convolution layers in CNN architecture is common. This affords a new type of invariance to translation.

Here the pooling layer works on each input depth slice separately by resizing it spatially. The most prevalent type is a size 2 filter pooling layer functioned with having 2 as stride, pooling at each depth slice by 2 in the input results in 75% of the activations removal.

Final layers : After the series of convolutional, ReLU and max pooling layer, the matrix is compressed into vector form which is fed to fully connected layer having the outputs equal to number of classes (categories) followed by soft max layer. This "loss layer" insists on exactly how training deal with the deviation between the expected (output) and true labels and is typically the neural network's final layer.

Softmax loss has been used here to calculate one class from K mutually exclusive classes. Sigmoid cross-entropy loss is used in [0, 1] to predict K autonomous likelihood values. Again the Euclidean loss is for returning the labels that are properly valued.

After completion of one forward pass, the back propagation operation starts in order to adjust the weight and biases of the processing elements (neurons) to reduce the loss. A neural network is trained by choosing all neuron weights so that the network learns from known inputs to estimate the target outputs. It's hard to logically fix a multi-layer network's neuronal weights. But the back-propagation algorithm offers an iterative solution for easy and efficient weight resolution. As an optimization technique, the typical version utilizes gradient descent. Gradient descent is relatively time-consuming and also not guaranteed to discover the slightest error, but it works well with adequate setup (recognized as hyper parameters in machine learning).

Layers specifications: The table (1) shows the parameters of the different layers in the SCNN. 15x1 arrays of layers are used in SCNN.

Table 1 CNN layers specifications

Type of the layer	*No. of Filters*	*Size/Stride*
Input	32 x 32 x 3	
Convolutional	32	5 x 5/1
Max pooling	1	3 x 3/2
Convolutional	32	5 x 5/1
Max pooling	1	3 x 3/2
Convolutional	64	5 x 5/1
Max pooling	1	3 x 3/2
Fully connected	64	
Fully connected	5	
Loss function	5 class scores	
Classification layer	Cross entropy	

This work is implemented in MATLAB.

Processing steps

- Use .mat files to label the images and to obtain the training data as well as test data.
- Define the layers to build a Convolutional Neural Network
- Train CNN with the trained data.
- Execute the network on the testing data.

Obtain classification Accuracy.

IV. SYSTEM SPECIFICATIONS

Processor: Intel(R) Core(TM) i5-6500 CPU @3.20GHz 3.19GHz

RAM: 8.00 GB

V. EXPERIMENTAL RESULTS

Dataset1: Object classification is performed by using two datasets, Dataset1: CIFAR10dataset [9] having 10 categories of general image classification with a total of 60000 images. These images are 32x32 color images with good quality. The image categories are Airplane, Automobile, Bird, Cat, Dear, Dog, Frog, Horse, Ship and Truck.

Dataset2: For Vehicle classification experiment with SCNN MIO-TCD [11] is used. Out of 11 categories available in the dataset, 5 categories are selected, having 600 images in each category which are multi view low resolution images.

Table 2 Input to CNN

Class name	*Number of samples*	
	Training	**Testing**
Car	500	100
Bus	500	100
Truck	500	100
Motorcycle	500	100
Van	500	100

Result Analysis:

The CNN framework is implemented on the two datasets. Table 3 shows results when the proposed method is applied on CIFAR10 dataset having 60000 images & on MIO-TCD database with 3000 low quality, multiple view images. Evidently when network is trained and tested on huge database by avoiding the over fitting, though the time factor is high due to implementation on single CPU, accuracy is acceptable.

In the case of MIO-TCD dataset for vehicle type classification, the accuracy can be uplifted when more training is done on the network. Also when good image enrichment methods are applied on these low quality input images results can be improved.

Table 3 Classification results when datasets applied on network

Parameters	*CIFAR10 Dataset*	*MIO-TCD Dataset*
No. of training images	50,000	2,500
No. of testing images	10,000	500
Max Epochs	10	15
No. of iterations	3900	750
Time Elapsed	42 hrs.	0.72 hrs.
Accuracy	73.71%	96.3%

VI. CONCLUSION

The results show that using supervised learning, CNN is capable of achieving good results with highly challenging datasets like MIO-TCD with low resolution images. Additionally the recognition and classification is viewpoint invariant. The network training options should be precisely selected to avoid overfitting or underfitting which could lead to accuracy degradation, observed with CIFAR10 dataset.

REFERENCES

1. Robert.Haralick , K.Shanmugam and Its'Hak Dinstein ,1973. Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics Vol: SMC 3 , Issue: 6 , pp. 610 – 621
2. Erin J. Bredensteiner, Kristin P. Bennett, 1999. Multi category Classification by Support Vector Machines Computational Optimization, pp 53-79
3. Olivier Chapelle, Patrick Haffner, and Vladimir N. Vapnik ,1999 .Support Vector Machines for Histogram-Based Image Classification. IEEE transactions on neural networks, vol. 10, No. 5, Pp. 1055-1064
4. Vailaya ,M.A.T. Figueiredo , A.K. Jain , Hong-Jiang Zhang, 2001. Image classification for content-based indexing. IEEE Transactions on Image Processing Vol: 10 , Issue: 1 pp. 117-130
5. Ya-ChunCheng ,Shu-YuanChen , 2003. Image classification using color, texture and regions. Image and Vision Computing Vol 21, Issue 9, 1 Pp. 759-776
6. G.M. Foody , A. Mathur, 2004. A relative evaluation of multiclass image classification by support vector machines. IEEE Transactions on Geoscience and Remote Sensing , Vol: 42 , Issue: 6 , pp. 1335-1343
7. G. Camps-Valls , L. Bruzzone , 2005. Kernel-based methods for hyper spectral image classification. IEEE Transactions on Geoscience and Remote Sensing. Vol: 43 , Issue: 6 , Pp.1351 - 1362
8. Tsung-Han Chan , Kui Jia , Shenghua Gao , Jiwen Lu , Zinan Zeng , Yi Ma , 2015. PCANet: A Simple Deep Learning Baseline for Image Classification? IEEE Transactions on Image Processing . Vol: 24 , Issue: 12 , pp. 5017-5032.
9. Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, Wei Xu, 2016. CNN-RNN: A Unified Framework for Multi-Label Image Classification. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2285-2294
10. Krizhevsky, A., Hinton. G., 2009. Learning multiple layers of features from tiny images. Master's Thesis, University of Toronto, Toronto, Canada.