International Journal of Scientific Research in Computer Science and Engineering
Vol.6, Issue.3, pp.22-26 , June (2018)

Research Paper

E-ISSN: 2320-7639

Image Classification Using Convolutional Neural Network

N.S. Lele

Dept. of Computer Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India

Available online at: www.isroset.org

Received: 26/May/2018, Revised: 06/Jun/2018, Accepted: 15/Jun/2018, Online: 30/Apr/ 2018

***Abstract*—** Image recognition, in the context of machine vision, is the ability of the software to identify objects, places, people, writing and actions in images. Computers can use machine vision technologies in combination with a camera and artificial intelligence software to achieve the task of image recognition. Image recognition is used to perform a large number of machine-based visual tasks, such as labeling the contents of images, performing image content search for guiding autonomous robots, self-driving cars and accidental avoidance system. While human brains recognize objects easily, computers have difficulty with the task. Software for image recognition requires deep machine learning. Performance is based on the complexity of convolutional neural network as the specific task requires massive amount of computational power for its computer-intensive nature. This work will review 'CIFAR-10' dataset which has classified images in various groups. This problem is a supervised learning task which will be able to classify any new images put forward from these various groups. This work also attempts to provide an insight into 'You Only Look Once (YOLO)' which is an example of unsupervised image classification. It can immediately classify the images into various objects by drawing rounded boxes around them and naming those objects.

***Keywords*—** Deep Learning, Convolutional Neural Network, Image Classification, Computer Vision

I. INTRODUCTION

The intent of the classification process is to categorize all pixels in a digital image into one of several land cover classes, or "themes". This categorized data may then be used to produce thematic maps of the land cover present in an image. Normally, multispectral data are used to perform the classification and, indeed, the spectral pattern present within the data for each pixel is used as the numerical basis for categorization. The objective of image classification is to identify and portray, as a unique gray level (or color), the features occurring in an image in terms of the object or type of land cover these features actually represent on the ground.

Rest of the paper is organized as follows, Section I gives the introduction of image classification, Section II gives the related work in image classification field, Section III describes the types of image classification, Section IV describes the convolutional neural network , Section V describes the various layers of convolutional neural network, Section VI describes the experiments of Cifar-10 dataset as well as YOLO (You Only Look Once), Section VII describes the methodology used, Section VIII describes the results of the experiments, Section IX describes the conclusion and future scope in image classification

II. RELATED WORK

Neural networks are basically built on the concept of biological neurons. There was an attempt to simulate this process by Warren McCulloch and Walter Pitts in 1943. In 2012, a paper from University of Toronto called " ImageNet Classification with Deep Convolutional Networks " was published, which went on to become one of the most influential papers in this field.

III. TYPES OF IMAGE CLASSIFICATION

Image classification is perhaps the most important part of digital image analysis. It is very nice to have an image, showing a magnitude of colors illustrating various features of the underlying terrain, but it is quite useless unless to know what the colors mean. Two main classification methods are Supervised Classification and Unsupervised Classification.

A. Supervised Image Classification

With supervised classification, we identify examples of the Information classes of interest in the image. These are called "training sets". The image processing software system is then used to develop a statistical characterization of the reflectance for each information class. This stage is often called "signature analysis" and may involve developing a characterization as simple as the mean or the rage of reflectance on each bands, or as complex as detailed analyses of the mean, variances and covariance over all bands. Once a statistical characterization has been achieved for each information class, the image is then classified by examining

the reflectance for each pixel and making a decision about which of the signatures it resembles most.

B. Unsupervised Image Classification

Unsupervised classification algorithms do not compare points to be classified with training data. Rather, unsupervised algorithms examine a large number of unknown data vectors and divide them into classes based on properties inherent to the data themselves. The classes that result stem from differences observed in the data. In particular, use is made of the notion that data vectors within a class should be in some sense mutually close together in the measurement space, whereas data vectors in different classes should be comparatively well separated. If the components of the data vectors represent the responses in different spectral bands, the resulting classes might be referred to as spectral classes, as opposed to information classes, which represent the ground cover types of interest to the analyst. The two types of classes described above, information classes and spectral classes, may not exactly correspond to each other.

IV. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks are very similar to ordinary neural networks; they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw images on one end to class scores at the other. They have a loss function in order to minimize the loss.

There are various advantages of using a convolutional neural network:

A. Sparse Representations

Let us assume that you are working on an image classification problem that involves the analysis of large pictures that are millions of pixels in size. A traditional neural network will model the knowledge using matrix multiplication operations that involve every input and every parameter which results easily in tens of billions of computations. Well, it turns out that the kernel in convolution functions tends to be drastically smaller than the input which simplifies the number of computations required to train the model or to make predictions. The result could be a few billion operations smaller and more efficient than traditional fully-connected neural network.

B. Parameter Sharing

Another important optimization technique used in Convolutional neural network is known as parameter sharing. Conceptually, parameter sharing simply refers to the fact that they tend to reuse the same parameters across different functions in the deep neural network. More specifically, parameter sharing entails that the weight parameters will be used on every position of the input which will allow the model to learn a single set of weights once instead of a different set for every function. Parameter sharing in Convolutional neural network typically results on massive savings in memory compared to traditional models.

C. Equivariance

Equivariance is a property that can be seen as a specific type of parameter sharing. Conceptually, a function can be considered equivariance if, upon a change in the input, a similar change is reflected in the output. Using a mathematically nomenclature, a function f(x) is considered equivariant to a function g() if f(g(x))= g(f(x)). It turns out that convolutions are equivariant to many data transformation operations which means that we can predict how specific changes in the input will be reflected in the output.

V. LAYERS OF CONVOLUTIONAL NEURAL NETWORK

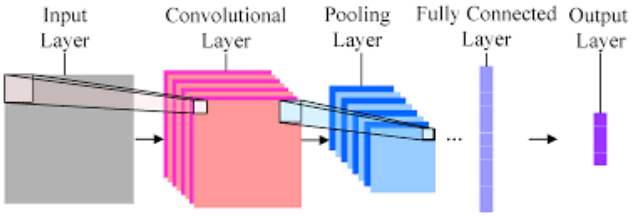


Figure 1: Convolutional Neural Network Layers

A. INPUT LAYER

The main task of this layer is to hold the input images. For example, if the image is of size 32*32*3, then it will hold raw pixel values of width 32, height 32, and with three color channels R, G, B.

A. CONVOLUTION LAYER

The main task of the convolutional layer is to detect local conjunctions of features from the previous layer and mapping their appearance to a feature map. As a result of convolution in neuronal networks, the image is split into perceptrons, creating local receptive fields and finally compressing the perceptrons in feature maps. Thus, this map stores the information where the feature occurs in the image and how well it corresponds to the filter. Hence, each filter is trained spatial in regard to the position in the volume it is applied to.

C. POOLING LAYER

The pooling or down sampling layer is responsible for reducing the spacial size of the activation maps. In general, they are used after multiple stages of other layers (i.e. convolutional and non-linearity layers) in order to reduce the computational requirements progressively through the network as well as minimizing the likelihood of over fitting.

D. FULLY CONNECTED LAYER

The fully connected layers in a convolutional network are practically a multilayer perceptron (generally a two or three layer MLP) that aims to map the activation volume from the combination of previous different layers into a class probability distribution. Thus, the output layer of the multilayer perceptron will have m*(l−i) outputs, i.e. output neurons where i denotes the number of layers in the multilayer perceptron.

E. OUTPUT LAYER

After multiple layers of convolution and padding, we would need the output in the form of a class. The convolution and pooling layers would only be able to extract features and reduce the number of parameters from the original images. However, to generate the final output we need to apply a fully connected layer to generate an output equal to the number of classes we need. It becomes tough to reach that number with just the convolution layers. Convolution layers generate 3D activation maps while we just need the output as whether or not an image belongs to a particular class. The output layer has a loss function like categorical cross-entropy, to compute the error in prediction. Once the forward pass is complete the back propagation begins to update the weight and biases for error and loss reduction.

VI. EXPERIMENTS

A. Cifar-10

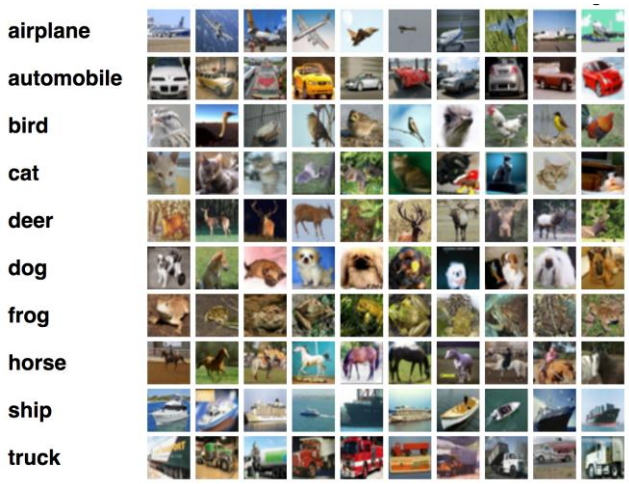


Figure 2: Cifar-10 classes

The CIFAR-10 dataset consists of 60000 32*32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

B. YOLO (YOU ONLY LOOK ONCE)

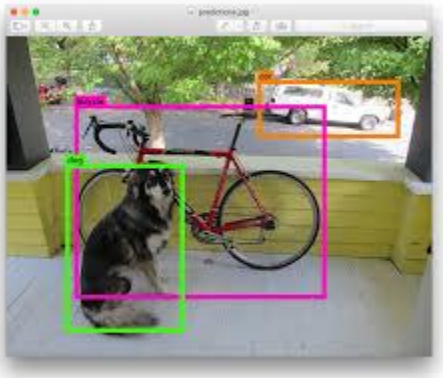


Figure 3: YOLO detection system[5]

Compared to other region proposal classification networks (fast RCNN) which perform detection on various region proposals and thus end up performing prediction multiple times for various regions in a image, Yolo architecture is more like FCNN (fully convolutional neural network) and passes the image (n*n) once through the FCNN and output is (m*m) prediction. Thus, the architecture is splitting the input image in m*m grid and for each grid generation 2 bounding boxes and class probabilities for those bounding boxes. Note that bounding box is more likely to be larger than the grid itself.

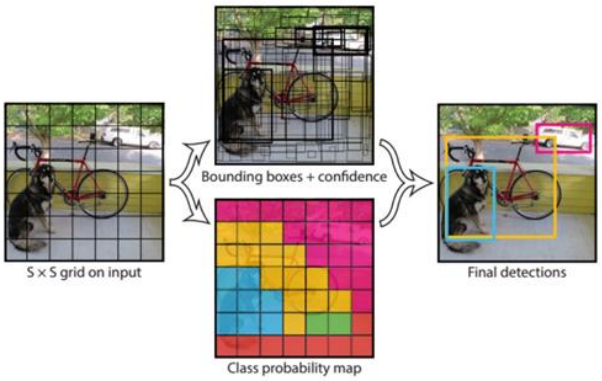
VII. METHODOLOGY

A. Cifar-10

Suppose that we have some N*N square neuron layer which is followed by our convolutional layer. If we use an m*m filter ω, our convolutional layer output will be of size (N−m+1)*(N−m+1). In order to compute the pre-nonlinearity input to some unit xℓij in our layer, we need to sum up the contributions (weighted by the filter components) from the previous layer cells.

The max-pooling layers are quite simple, and do no learning themselves. They simply take some k*k region and output a single value, which is the maximum in that region. For instance, if their input layer is a N*N layer, they will then output a N/k*N/k layer, as each k*k block is reduced to just a single value via the max function.

B. YOLO (YOU ONLY LOOK ONCE)



The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

Figure 4: YOLO Classification Process[5]

The network has 24 convolutional layers followed by 2 fully connected layers. 1*1 reduction layer is followed by 3*3 convolution layer. We optimize for sum-squared error in the output of our model. Localization error is treated on the same lines as classification error which may not be ideal. Many grid cells may not contain a part of the image, which can lead to decrease in confidence scores. In order to remedy this, we increase the loss from bounding box coordinate predictions and decrease the loss from confidence predictions for boxes that do not contain objects. For this, two parameters λcoord = 5 and λnoobj =0.5.

Sum-squared error also equally weights errors in large boxes and small boxes. Our error metric should reflect that small deviations in large boxes matter less than in small boxes. To partially address this we predict the square root of the bounding box width and height instead of the width and height directly.

VIII. RESULTS

A. Cifar-10

TABLE 1: ACCURACY OF CIFAR-10

Iterations	4-Layer CNN	6-Layer CNN
10	71.29%	75.61%
20	74.57%	75.31%
100	67.06%	68.66%

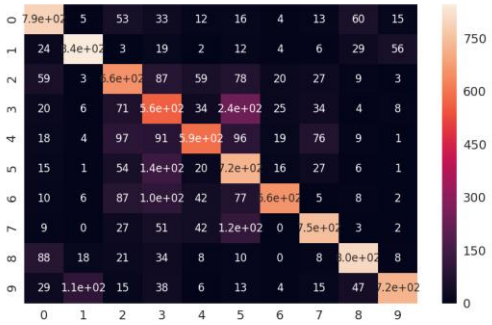


Figure 5: Confusion matrix for 6-layer CNN

B. YOLO (You Only Look Once)

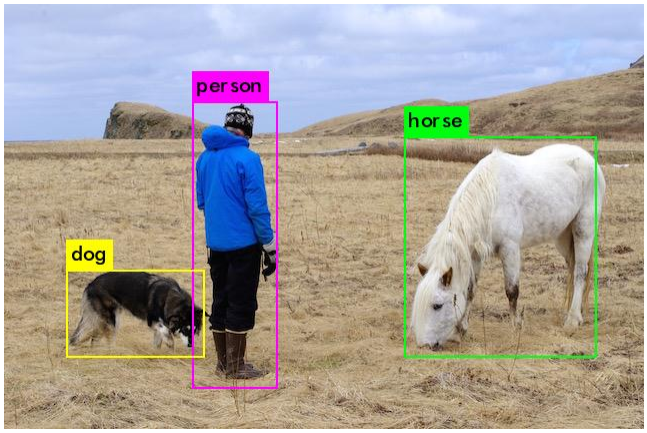


Figure 6: Classified Image

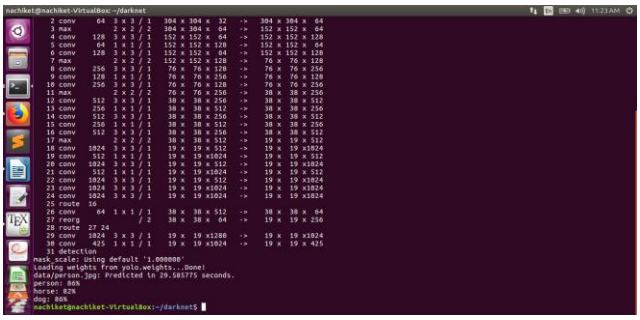


Figure 7: Confidence percentage from YOLO

Table 2: Confidence percentage of classified image

Sr. No.	Object	Confidence
1	Person	86%
2	Horse	82%
3	Dog	86%

IX. CONCLUSION AND FUTURE SCOPE

This work aims at classifying images using Convolutional Neural Network (CNN). With the optimization possible with CNN, it is easier to classify images as compared to traditional image classification algorithms. With further enhancement in study of neural networks, image classification problems will continue to become more and more easier to solve. With image classification finding applications in various spheres of life, neural networks have assumed even more significance. In future, this work can be extended for real time image processing in various fields like validation and verification of different real time images, spoofing.

ACKNOWLEDGMENT

N.S.Lele would like to thank the Head of Department of Computer Engineering PICT, Pune in providing the opportunity to compile this paper. N.S.Lele is also thankful to his guide Prof. Sumit Shinde for his guidance in writing this survey paper.

REFERENCES

[1] Chan T H, Jia K, Gao S, et al. "*PCANet: A simple deep learning baseline for image classification*," arXiv preprint arXiv:1404.3606, 2014.

[2] TKrizhevsky A, Sutskever I, Hinton G E, "*Imagenet classification with deep convolutional neural networks*," Advances in neural information processing systems, pp. 1097-1105, 2012**.**

[3] Bouvrie J, "*Notes on convolutional neural networks*," Neural Nets, 2006.

[4] Chan T H, Jia K, Gao S, et al. "*PCANet: A simple deep learning baseline for image classification*," arXiv preprint arXiv:1404.3606, 2014.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "*YouOnlyLookOnce: Unified,Real-TimeObjectDetection,*" arXiv:1506.02640[cs.CV]

Authors Profile

Mr. N S Lele is currently pursuing Bachelor of Computer Engineering from Savitribai Phule Pune University. His main research work focuses on Machine Learning and Artificial Intelligence.

