Convolution Neural Network Based Image Classifier

Hussain Qudsia Ejaz
{hejaz2@asu.edu}
Department of Computer Science, Jamia Hamdard

Syed Ali Mehdi
{alimehdi@jamiahamdard.ac.in}
Department of Computer Science, Jamia Hamdard

Abstract. The concept of Deep Learning is emanated in machine learning as an enhanced research area and is empirical to various image applications. The objective of the project propounded in the paper, is applying the abstract of an algorithm of Deep Learning, viz, Convolutional neural networks (CNN) for multiple image classification. The algorithm is assessed on variegated datasets, which consist 2399 images taken from google, myntra fashion clothes, etc. The algorithm's performance is gauged based on the quality metric known as Confusion Matrix. The analysis is done and the model successfully classifies each image using VGG19 model of CNN.

Keywords: Multiple predictions, image classification, confusion matrix, pyramid reduction

1. Introduction

Image processing is a boundless subject and presuppose processes that are complex mathematically [1]. Image processing encompasses some quintessential operations namely image rectification, image enhancement, image fusion, images classification etc. Image classification forms a foremost part of image processing. The purpose of image classification is the self-allocation of image to abstract classes [1]. There are two types of classification, unsupervised classification and supervised classification. The process involves two steps, training of the model followed by testing. The training technique means, to take the idiosyncratic properties of the images and form an eccentric description for a specific class. Depending on the type of classification problem, this process is done for all the class; binary classification or multi-class classification. The next step which is testing, means to sort these test images under those classes for which the system was trained. Assigning of class is done according to the partitioning within the classes with respect to the training features. Deep structured learning, since 2006 has more customarily called deep learning, as a new and emerged research area of machine learning [2]. There are various definitions available for Deep Learning; one of these definitions from [2]. A class of machine learning system that wring layers of the nonlinear information processing for unsupervised or supervised transformation of the feature extract for classification and pattern analysis is elucidated as Deep Learning. This project is one of the application of Convolutional Neural Network or CNN for multiple image classification. The image data used for testing the algorithm includes data from google, myntra fashion blog, amazon fashion blog. The remaining paper is organized as follows: The next section that is, Section 2, of the paper explains working of the

ICIDSSD 2020, February 27-28, New Delhi, India

DOI 10.4108/eai.27-2-2020.2303205

network and theoretical background. The procedure is detailed in Section 3. The next section, Section 4 deals with the observation recorded using CNN.

2. Image classification

Image classification means classifying an image into heterogeneous classes or categories. There are many provocation while classifying an image. Some often confuses with the context of image classification and that with the object detection, for an example see **Figure 1**. The classification of an image into a certain category uses image classification. Image classification can also be used when classification is done into an irrefutable category. If the location of objects should be known as well, object detection can be used.

Fig. 1. Difference between classification and object detection.

3. Convolution Neural Network

Mathematical models of the neural networks have been using since ages, the very first model propounded was by McCulloch and Pitts in [3]. Neural networks are the networks that are made up of a number of layers with each of the layer connected to the other layers. A feed-forward neural network or FFNN which is also known as neural activation of the connections between the pairs of neurons [4] In FFNN, the neurons are anchored in a directed way having clear initialization and stop place which is also named as the input layer and the former as output layer. Hidden layer is the intermediate layer. Adjustment of weights is done and the minimization of error between the output obtained and the input that goes into the input layer is achieved by the process of learning. The weights are adjusted by a method known as back propagation (that is partial derivation of the error which is obtained with respect to previous layers of weights). The process of weight adjustment is recapitulated in an inductive manner until weight layer which is connected to input layer is improved.

Convolutional Neural Networks (CNN) is derivatives of Multi- Layer Perceptron (MLPs) which are induced from the biology. The filters used are local in input given and are therefore, better to wring the structural local interconnection that exists in images that are natural. [5]. Convolutional neural networks are delineated to process the images which are two-dimensional (2-D) image [6]. A CNN architecture used in this project is that defined in [11]. The project uses VGG19 model which is directed on millions of images that are in ImageNet database.

3.1 Working of CNN Algorithm

The detailed explanation of working of the algorithm is present in [7]. The 2D image is given as an input to the neural network. The network consists of the layers which takes the

image as an input, output layer are those layers from where we get the trained output and the intermediary layers that is the hidden layers. This network has a succession of convolutional and sub-sampling layers. Together, the layers assembles a correspondence of input image data's. CNNs exploit extensively local interrelation by compelling a local connecting patterns in allying the neurons of abutting layers [8]. Neurons in layer 'm', are connected to previous layer's subset neurons of (m-1), where the neurons of the (m-1) layer have conterminous receptive fields. In this project, VGG19 model is used. This network is 19 layers deep, with learnable weights, which consist of sixteen convolution layers and three fully connected layers. **Figure 2** describes the layers about VGG19.

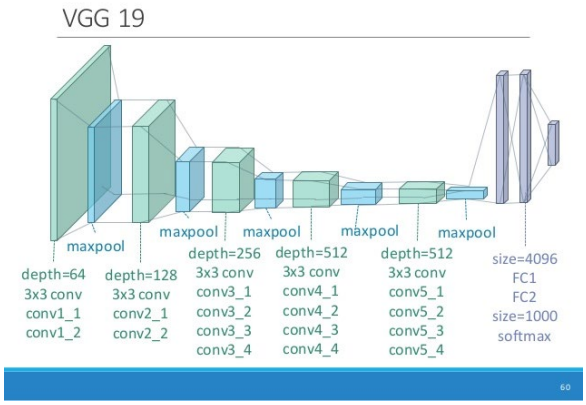

Fig. 2. VGG19 model [13]

4. Challenges faced

The function of agnizing a concept (e.g. cat) is inconsequential for a human being to implement. In view of the challenges, it is worth to convolute it from the perspective of a Computer Vision algorithm.

The list of these problems below is provided as a 3-D series of brightness values:

• **Variation in viewpoint**: the orientation of a single instance of an object can be achieved in many ways depending on the camera.

• **Size variation**: Visual groups frequently show differences in sizes (size in the real world, not its size in images).

• **Deformation**: Many engrossment objects are not invariable bodies and can be deformed in many ways.

• **Occlusion**: You can wedge certain objects of interest. Sometimes, the brightness can be as low as few pixels.

• **Illumination condition**: The pixel level is moving with the effect of the illumination.

• **Clutter in the background:** These objects can blend into their surroundings, making them hard to identify.

•**Intra -lass variation**: Engrossment groups, for example chair, can often be relatively broad. These objects are of different types, each with its own aura.

An approved model for the classification of images must be permanent to the cross product of all the above variations, while at the same time perpetuating sensitivity to the variations between classes.

5. Data-driven approach.

The most important question here is generating an algorithm which classifies all the images into different classes. One of the example can be sorted list of the numbers. This will obviously be different form an algorithm that is written to identify cats in an image. In data-driven approach, the computer is provided with several types of example depending upon the categories. Hence, the code does not require the class of interest mention for each category. Instead, this is replaced by the various data of each category. The learning algorithm is then developed by the class which looks at the provided examples and then the visual appearance is learned by the algorithm of those classes.

Here's the key stages that machines uses to identify patterns in an image:

Convolution: An image is negotiated to identify explicit features in an image. Convergence the helps in blurring, edge detection, sharper, noise reduction, etc. on an image which can help the machine learn the specific characteristics of an image.

Pooling: Convoluted images needs to be reduced since it can be too large. Pooling is done to minimize the image without untethering features or patterns.

Flattening: It transforms a two-dimensional (2-D) matrix of features to a vector of features that is provided to feed into a neural network.

Full-Connection: This refers to the process of fortifying the flattened image to a neural network.

6. Methodology

This project is based on classification of clothes according to the type of clothes, that is, men clothes, women clothes and shoes. The main objective is to return multiple predictions using deep neural network. This was achieved by using VGG19 model of CNN, and the accuracy was measured by confusion matrix. The detailed explanation of this model can be found in [18]. The model successfully classified images into specified classes using number of epochs. This project achieved the accuracy of 89% in two epochs. The programming language used in this project is python.

7. Procedure of the Experiment

7.1 The Data Set

An image is provided as an input which is ten transmuted in the grey scale since the information of data of the images is predominant. This data is used for the network. The coloured information of the images is not as such important. The images are then rescaled to the size, 32x32. Pyramid reduction technique is used to reduce the size of the images since the dataset images are quite large. The data structure that is used is the image pyramid which is formulated to support the adeptness of the scaled convolution throughout image representation. This data structure, consists a sequential copies of an original image where the sample density as well as the resolution is decreased regularly [9].

The Data set includes various categories of clothes. The data is divided into two sets: training data sets and testing data sets. The larger portion of the above data is sent for training and small portion is sent for testing.
The Data set we will be using is the collection of data of various types of clothes.
The dataset of this project contains **2,399** images in **eight classes**, which includes:

- Jeans of men (322)
- Women dress (378)
- Pants (301)
- Shirt (276)
- Shoes (410)
- Women pant (386)
- Women top (326)
- Women shoe(410)

7.2 Data Training

The purpose of training algorithm as in [7] is to train a network so that the error between the output of the network and the desired output is minimised. The error function is derived from the following equation (1) and remains the same for all weights as well as bias terms.

$$E(w) = \frac{1}{K \times N} \sum_{k=1}^{K} \sum_{n=1}^{N_L} (y_n^k - d_n^k)^2 \quad (1)$$

The gradient of error is enumerated by sensitivities of error, which are elucidates as the partial derivatives of the error function in accordance with the weighted total input to a neuron. Once the error gradient has been procured, numerous optimization algorithms can be adjusted to train the network to curtail the energy function.

Firstly, the data set is used to train the algorithm known as the training data, see **Figure 3**. Most of the data is used for preparation, and a smaller amount is used for research than that of training data.

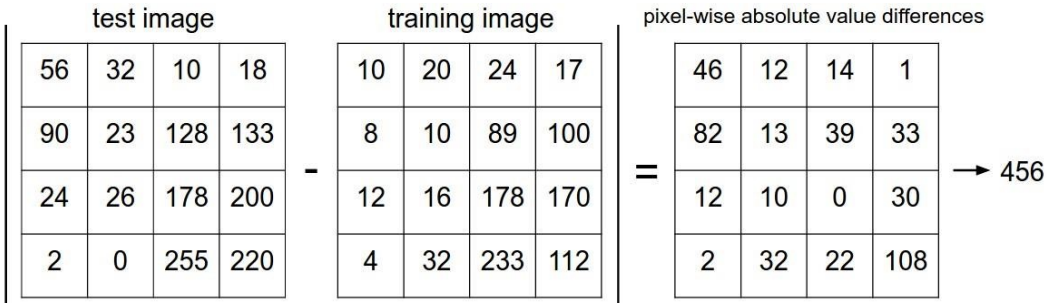

Fig. 3. Definition of using pixel-wise differences that equate two images to L1 distance (in the definition above for a single colour channel). Two images are subtracted element-wise and a single number is applied to all variations. If the two images are similar, the consequence is zero. But if the images are different then the result is big [15].

A good model of image classification must be invariant to the cross product of all these variations, while at the same time maintaining sensitivity to the variations between groups.

7.3 Load the data set

To load data sets out of the box, a library called datasets is used. The data can be downloaded from the server which speeds up the process as downloading the data from the computer may take longer time. The data is downloaded from google in this project, and is stored in the folder called dataset. Together with the labels the train and test images are loaded and stored in the variables Tr_x, Tr_y, Te_x, Te_y. The training data in this project has a form of 60000 x 28 x 28 since there are 60,000 training samples of 28 x 28 measurements each. Similarly the test data has a shape of 10,000 x 28 x 28 since 10,000 test samples are available, see **Figure 4.**

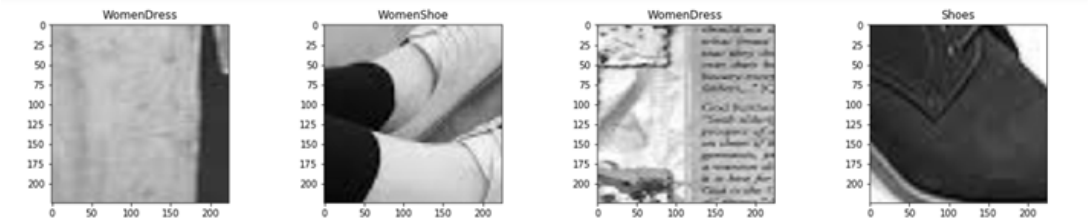

Fig. 4. Illustration of the data set that is loaded

7.4 Load the model based on VGG19

In the code of this project, first line will load the VGG19 model. It may take more than the required time. By executing second line, the summary of the existing model can be seen. It has a lot of convolutional, pooling and dense layers. Executing third line, this model is of type "Model". In next step, there is creation of a model of type "Sequential". After this the layers needs to be freeze so that there is no change to any weights or biases associated with the layers. Hence, freezing of layers is done. These layers is used as all these layers are already well trained on image classification problem.

7.5 Neural network architecture

The Convolutionary layer with Conv2D) (is applied to this project. Remember that this feature is used because photographs are used for the job. Next, the activation function Leaky ReLU is added that helps the network learn the boundaries of nonlinear decisions. Since there are ten different classes, there is a need for non-linear boundary decisions that could separate those ten classes that are not linearly separable.

More precisely, Leaky ReLUs os added because they are trying to fix the dying Rectified Linear Units (ReLUs) problem. First, MaxPooling2D) (adds the max-pooling line, and so on. The last layer is a Dense layer which has a 10-unit softmax activation function necessary for this multi-class classification problem, see **Figure 5**.

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (17): ReLU(inplace=True)
    (18): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (19): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (22): ReLU(inplace=True)
    (23): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (24): ReLU(inplace=True)
    (25): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (26): ReLU(inplace=True)
    (27): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Fig. 5. Neural Network Architecture

7.6 Train the model

Two epochs are required to train the model. The history of the object will be return by a function fit(). The accuracy as well as the loss function that is plotted between training and the validation, helps in the perfect analysis of the performance of model. This is accomplished by the result of the fit() function that is stored in the form of result. The model is trained according to this function. The accuracy achieved by the model was around 90% and the training loss accordingly was quite low.

7.7 Prediction of classes

This section presents the result of the classification obtained using CNN algorithm on the dataset provided. The results are presented which classifies the image into specific classes, see **Figure 6**. The result also shows the predicted result and the truth result. The accuracy achieved is good and acceptable.

Fig. 6. The predicted classes

8. Conclusion

This project is an example of implementing image classification that provided some valuable information, which can be applied to future projects. Some of which can be, data acquisition, transfer learning and model evaluation. Since the project predicted the type of clothes, this project can be used in the places where the users can upload the picture and the system will automatically provide similar sets of clothes images or the links that can be used to purchase similar dresses online. Note that the code can be easily modified to allow for multiple classifications. There is, however, a problem which must be considered. For each combination of categories that would like to be predicted, the training data; just as a neural network cannot predict classes it has never been trained on. The neural network cannot predict multiple class labels for combinations it has never seen. The reason for this behaviour is due to neuronal activations inside the network. It seems there were a lot of parameters to choose from and then it took a long time for the training. These two issues shouldn't stop implementing this particularly when the work is on simple issues. Most researchers working in this field are kindly open-sourcing their trained models that have been trained on millions of images, and on many GPUs for hundreds of hours. Rather than starting from scratch, their templates can be used as the starting point.

References

[1] Lillesand, T.M. and Kiefer, R.W. and Chipman, J.W., in “Remote Sensing and Image Interpretation” 5th ed. Wiley, 2004

[2] Li Deng and Dong Yu "Deep Learning: methods and applications" by Microsoft research [Online] available at: http://research.microsoft.com/pubs/209355/NOW-Book-RevisedFeb2014-online.pdf
[3] McCulloch, Warren; Walter Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics 5
[4] An introduction to convolutional neural networks [Online]available at:http://white.stanford.edu/teach/index.php/An_Introduction_to _Convolutional_Neural_Networks
[5] Hubel, D. and Wiesel, T. (1968). Receptive fields and functional architecture of monkey striate cortex. Journal of Physiology (London), 195, 215–243C. J. Kaufman, Rocky Mountain Research Laboratories, Boulder, Colo., personal communication, 1992. (Personal communication)
[6] Yann LeCun, Leon Bottou, Yodhua Bengio and Patrick Haffner, "Gradient -Based Learning Applied to Document Recognition", Proc. Of IEEE, November 1998.
[7] S. L. Phung and A. Bouzerdoum,"MATLAB library for convolutional neural network," Technical Report, ICT Research Institute, Visual and Audio Signal Processing Laboratory, University of Wollongong. Available at: http://www.uow.edu.au/˜phung
[8] Tutorial on deep learning [Online] available at : http://deeplearning.net/tutorial/lenet.html

[9] Adelson, Edward H., Charles H. Anderson, James R. Bergen, Peter J. Burt, and Joan M. Ogden. "Pyramid methods in image processing." RCA engineer 29, no. 6 (1984): 33-41.

[10] M. Riedmiller and H. Braun, "A direct adaptive method of faster backpropagation learning: The rprop algorithm", in IEEE International Conference on Neural Networks, San Francisco, 1993, pp. 586– 591.
[11] Karen Simonyan∗ & Andrew Zisserman+ Visual Geometry Group, Department of Engineering Science, University of Oxford {karen,az}@robots.ox.ac.uk
[12] https://www.hackerearth.com/blog/developers/introduction-to-object-detection/

[13] https://towardsdatascience.com/transfer-learning-in-tensorflow-9e4f7eae3bb4

[14] https://www.pyimagesearch.com/2018/05/07/multi-label-classification-withkeras/

[15] https://towardsdatascience.com/object-detection-using-deep-learning-approaches-anend-to-end-theoretical-perspective-4ca27eee8a9a

[16] https://tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-ofdeep-learning/ https://www.datacamp.com/community/tutorials/object-detection-guide

[17] http://cs231n.github.io/classification/

[18] https://analyticsindiamag.com/learn-image-classification-using-cnn-in-keras-withcode/ https://keras.rstudio.com/reference/application_vgg.html

[19] Very Deep Convolutional Networks for Large-Scale Image Recognition, K. Simonyan, A. Zisserman.

[20] https://www.datacamp.com/community/tutorials/convolutional-neural-networkspython

[21] Stack Overflow - Where Developers Learn, Share, & Build Careers

[22] VGG in TensorFlow

[23] https://theprofessionalspoint.blogspot.com/2019/07/fine-tune-vgg16-model-forimage.html

[24] https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/conceptobject-detection