Object Detection in Real Time Video

Shailendra Aote^{1*}, Pankhudi Tiwari², Manjiri Butolia³,
Aaishwarya Suradkar⁴ and Shrasti Gupta⁵

^{1,2,3,4,5}Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India

ABSTRACT

In this modern era, it is essential for various brands and businesses to have a technology that will be helpful for them with respect to the promotion of their business. Existing logo detection research focuses on closed set scenarios. The logo domain is too large for these strategies and requires an open set approach. In this paper, a logo detection and recognition system is proposed which detects the logos in the video. The issue of speed and size is addressed by proposing a compressed convolution neural network model VGG16 in Machine Learning. The proposed model improves on size, speed, and solves degradation issues. VGG16 trains the model and further classifies the images in order to detect the images. In comparison to VGG16 the proposed model is 88.4% smaller in size and 23.86% faster in training time. Thus, this model helps recognizing the images from the video with higher accuracy.

KEY WORDS: CONVOLUTION NEURAL NETWORK (CNN), MACHINE LEARNING, LOGO DETECTION, VGG-16.

INTRODUCTION

Nowadays visual content is considered to be the most powerful and informative among all the communication means that multinational and national brands all over the globe are using to showcase and advertise their brands. As the world is leaning towards digital platforms for almost anything and everything it becomes a beneficial and powerful means of advertising ones brand and catching eyes of millions of people in one short as well as to gain an edge over the competitors. For this very purpose owners invests money in online advertisement or even the digital events but the problem arises when the money invested for advertisement through online or digital means become worthless due to the issue of difficulty in coming up with the right amount of money

to be sponsored in any digital event, this issue occurs as there is uncertainty and dissatisfaction of whether the amount invested is worth the time for which brand logo is appearing on the screen throughout the event.

Logos are considered to be the mean of identification for the company, organization or institution throughout the globe. Logo is crucial for identity purpose and logo matching is one of the challenge in the recent era. There are some difficulties in logo matching such as different images and videos contains logo with different shapes and size also the location of logo is not fixed in each image. For the detection of logos in such a situation, traditional methods which includes the manual calculation fails. To overcome the problem associated with traditional method Logo Detection System comes up with an automated method which would count the total time period for which the logo appears in the video which will help brand owners decide the suitable amount to invest. For the deployment of this application in real-life scenario we would need to create an effective GUI so that even a lay man can interact with the application with almost no efforts. Along with this we will also be required to train our model on huge dataset as there are immense amount of brands around the world.

ARTICLE INFORMATION

Received 11th Oct 2020 Accepted after revision 24th Dec 2020
Print ISSN: 0974-6455 Online ISSN: 2321-4007 CODEN: BBRCBA

Thomson Reuters ISI Web of Science Clarivate Analytics USA and
Crossref Indexed Journal

NAAS Journal Score 2020 (4.31)
A Society of Science and Nature Publication,
Bhopal India 2020. All rights reserved.
Online Contents Available at: <http://www.bbrc.in/>
Doi: <http://dx.doi.org/10.21786/bbrc/13.14/23>

System Model

- We have designed a CNN model by using Python as a programming language.
- We used open source library in python called as TensorFlow which is widely used in today's world for machine learning applications such as neural network.
- Google Colab is used as a development environment.

In this paper we are working on the dataset of approximately 1000 images containing logos of different companies around the globe and using this dataset to develop an image classification tool which will characterize images in accordance with the company specific logo present in the frame of video by mapping each frame aspect to each image is the dataset provided.

METHODOLOGY

Initially we divided the process into three main categories which are: 1. Convolution Neural Network that is used to classify images by passing each image through the layers of CNN, 2. Implementing VGG-16 as a pretrained model to categorize frame of the input video and check accuracy for the same, 3. Calculating screen time for the respective company logo in the video.

A.Convolutional Neural Network: The convolutional neural network (CNN) is a part of deep learning neural networks that uses multiple preception that analyze image inputs and have learnable weights and bases to several parts of images and able to segregate each other. The advantage of CNN compared to other similar classes available is that it is self capable of detecting important features with any human intervention. CNN is also computationally efficient. It uses special convolution and pooling operations and performs parameter sharing.

a.Convolution Layer: The main purpose of the convolution step is to extract features from the input image. The convolutional layer is always the primary step during a CNN. We've an input image, a feature detector, and a feature map. The task is to take the filter and apply it pixel block by pixel block on the input image. We have to do this through the multiplication of the matrices. If we have a 2-Dimensional image input I , and a 2-Dimensional kernel filter K , the image convoluted is calculated as

$$(i,j) = \sum \sum (m,n) k(i-m, j-n)$$

b. Non-linear activation function(ReLU): ReLU applies an activation function onto the feature maps to extend the nonlinearity within the network. Because the images are highly non-linear. ReLU activation function performs a major task of discarding negative values from activation map by changing them to zero.

c. Pooling Layer: The disadvantage of the feature map output of convolutional layer is that it stores the exact position of features in the input which in result means that during the process of cropping, rotation or any

other minor changes to the input image will completely results in a different feature map. To beat this problem we use Pooling which is defined as to progressively reduce the spatial size of the representation to scale back the parameters and computation within the network. Pooling layer operates on each feature map independently. It makes it possible to detect objects in an image regardless of where they're located.

d. Fully Connected Layer: The main purpose of the artificial neural network is to combine our features with more attributes. This is done to make sure that the prediction of the classes are done with greater accuracy. This combines features and attributes which can predict classes more efficiently.

Figure 1: Element wise matrix multiplication and summation of the results onto feature map in convolutional layer.

B. VGG-16 Model: VGG-16 is the name of a pre-trained convolutional neural network invented by Simonyan and Zisserman from Visual Geometry Group (VGG) at University of Oxford in 2014 and it was able to be the 1st runner-up of the ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2014 in the classification task. VGG Net has been trained on ImageNet ILSVRC data set which include images of 1000 classes split into three sets of 1.3 million training images, 100,000 testing images and 50,000 validation images. 16 is employed with VGG because it uses 16 layers for filtering. The model obtained 92.7% test accuracy in ImageNet. Following are the 16 layers of VGG-16:

1. Convolution with 64 filters
2. Convolution with 64 filters along with Max pooling
3. Convolution with 128 filters
4. Convolution with 128 filters along with Max pooling
5. Convolution with 256 filters
6. Convolution with 256 filters
7. Convolution with 256 filters along with Max pooling
8. Convolution with 512 filters
9. Convolution with 512 filters

10. Convolution with 512 filters along with Max pooling
11. Convolution with 512 filters
12. Convolution with 512 filters
13. Convolution with 512 filters along with Max pooling
14. Fully connected with 4096 nodes
15. Fully connected with 4096 nodes
16. Output layer with Softmax activation with 1000 nodes.

C. Screen Time Calculation: We are giving 1000 images of different logos of the companies around the globe for training and validation which will be processed by CNN and trained by VGG-16. For the result and actual implementation of Logo Detection we are giving input in the form of mp4 video which includes logos of different brands. The video can be of any products advertisement or any online streaming event which is sponsored by some product companies along with the csv file for the respective video. As the final result we will iterate through the whole video and calculate the screen time of every brand in seconds.

IV. Dataset: The training archive contains cumulatively 1000 images of logos. Our objective is to build a robust image classification model with restrictions of consisting of few training samples of logos (multi-classification problem). We need to downscale the number of images to add a constraint on the input images. We trained our model for validation of approximately 1000 logo images. We have given labels to a few images and train the model on them. After learning patterns by the model, we can use it to predict an unseen set of images. We have mapped the csv file into two attributes, Image Id which consists of the name of each image along with the extension (.jpg), and class which consists of the corresponding class for each image or each logo type.

Figure 2: Dataset

We leverage the pre-trained VGG-16 model as a feature extractor and transfer low-level features, such as edges, corners, rotation and learn new level features specific to the target problem which is to classify the images. The model takes an input image of shape (224 X 224 X 3). After resizing and preprocessing the images we divide

Figure 3: Output

The screen time of Coke is 16 seconds
 The screen time of BMW is 0 seconds
 The screen time of Yahoo is 0 seconds
 The screen time of Google is 0 seconds

the dataset into training and validation set. Getting the features of the training and validation set we then use those features to retrain the model. Then we build the model followed by compiling and training the model.

CONCLUSION

This paper proposes a complete logo detection/recognition from a video. This software can calculate the total time of the advertisement displayed in the video. It makes the invested money in the advertisement worth the sponsorer and avoids dissatisfaction among them. It will help any company to decide the amount to pay for advertising, based on the number of times its logo is displayed in the advertisement. The results fetched for logo detection in a real-time video is an evidence to show that the overall performance of the system is suitable for practical use. The main objective of this paper is to give an in depth information about the Logo Detection System, the technology and algorithms running in the backend of the system and how the system as a whole can be proved beneficial for digital event sponsoring.

REFERENCES

- Andras Tuzk'o, Christian Herrmann, Daniel Manger, Jürgen Beyerer, "Open Set Logo Detection and Retrieval", October 2017
- Hussam Qassim Abhishek Verma David Feinzimer, "Compressed Residual-VGG16 CNN Model for Big Data PlacesImage Recognition"
- M Manoj Krishna, M Neelima, M Harshali, M Venu Gopala Rao, "Image classification using Deep learning" International Journal of Engineering & Technology, 2018.
- Steven C. H. Hoi, Xiongwei Wu, Hantang Liu, Yue Wu, Huiqiong Wang, Hui Xue, Qianf Wu, "LOGO-Net: Large-scale Deep Logo Detection and Brand Recognition with Deep Region-based Convolutional Networks", November 2015
- <https://www.learnopencv.com/keras-tutorial-transfer-learning-using-pre-trained-models/>
- <https://www.hackerearth.com/practice/machine-learning/transfer-learning/transfer-learning-intro/tutorial/>
- <https://iq.opengenus.org/vgg16/>
- <https://neurohive.io/en/popular-networks/vgg16/>