Object Recognition using Tensorflow and Convolutional Neural Network

Shreyansh Sharma[1], Harshit Panndey[2]

[1]*Student, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India*
[2]*Student, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India*

Abstract - *Our World is filled with a lot of visual information. As technology progresses the amount of knowledge that we can collect or obtain from images also increases. We can use the science of object recognition for many useful applications in order to enhance the knowledge obtained from digital data. We can use these techniques to train machines to act or react based on the captured visual data. For all these activities a basic necessity is to identify different type of objects in a given image. Considering the root of all we must first know how to identify a single object in a single Image and only then we can enhance the system to be more artificially intelligent. To identify a single Object in an Image is easy as it can be done easily by comparing spatial data of multiple images of similar type. Similarly multiple objects in an image can be recognized by using multiple different models for different type of objects. Here we are going to recognize single object in an image using an efficient repository Tensorflow. Using the proposed method we would obtain more efficiency in storing the data and will allow us to use the same data for multiple applications if needed.*

Key Words: Digital data, Machines, Artificially, Spatial Data, Tensorflow.

1. INTRODUCTION

Machine learning algorithms have revolutionised the way people see technology. From a simple machine that reduces human effort and help them in their respective tasks, they have turned into fully capable systems that can work independently and efficiently. Machines have become smarter, faster and efficient. From the very beginning we have tried to develop technologies that can help us to store and retrieve data. We have worked on different technological infrastructures that can help us to improve security and performance. And in this day and age where we have a lot going on around us at all times, it is very tough to keep track of all the data. We need more efficient ways to retrieve data that can help us in multiple ways, especially data we can get from inspecting the visual elements. Object detection is a way of detecting and labelling different objects in an image. It can be used in various fields, it can be used in autonomous vehicles for lane detection and road crack detection, it can be used in medical devices for detecting foreign cells and microorganisms, it can be used in security systems, tracking systems, detection systems, etc. There are simple and complex approaches for detecting static and dynamic objects in an image. There are ways to detect images based on the 3-D structure of the image, the depth field and image geometry [1]. Such methods are really efficient when dealing with simple image data with consistent information about the subject matter. Such methods can be used for simple object detection for e-appliances or simple mobile apps that can detect stationary objects. Object recognition is not just bound to detecting objects it can be modified and used for detecting human movements, emotions and it can even facilitate on a basic level an easy way for us to understand animals based on their body language [2]. Most of the modern object detection considers images as collection of points, where each point has some information associated to it. They are inspired from the pictorial structure models introduced by Fischler and Elschlager. The basic idea is to represent each part of the image as a collection of points where multiple sections of the image can be linked together based on the similarity of the properties of the data points [3]. There are various approaches that use point data such as colour properties in an image for detecting an object [4]. Object detection has been a part of computer vision applications and to facilitate the use of computer vision concepts and functions we have OpenCV library that is used in almost every object detection model and is very necessary at the basic level [5]. Using all the basic tools we can create detection tools that work for a particular label class of image features. The "features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection" [6]. There have been tremendous improvements in the field of machine learning and with it we have used developed specific neural networks made for object detection. We have developed various different convolutional network models which can be used for object detection. We have fast R-CNN and RPN networks that can be used for object detection. [7] In this paper they have shown the different types of network and how effective they are for object detection. Among these R-CNN have been improved and used for object detection in high quality image with a vast amount of data points [8]. After developing faster and more scalable R-CNN models we developed domain adaptive R-CNN. One of such model has been used for detection in the wild [9]. Even though we have such advance detection networks there still exist certain problems like object location detection in an image [10] and platform wise performance issues. There have been discussions about the multiple hierarchical object detection models on different platforms such as cortex [11] but that doesn't solve the predominantly existing issues. Here In this paper we have tried to check the performance of a simple object detection system based on simple image processing technique that uses tensorflow for storing information about the data points on an image. And that spatial information is then used for training the CNN to detect simple objects listed in the labels directory.

2. PROPOSED ARCHITECTURE

The proposed model has been divided into three modules

Image_Classifier Module: This is the module in which the model is created in a sequential manner layer by layer. The training data is also processed in this module and the trained data is then saved in a .h5 file.

Model_test: This module is use to test the training data and check for the test losses and test accuracy.

IR_tester: This module is responsible for getting user input from the user in the form of an object. This module is responsible for classifying the label for the image from the given set of labels. This module is responsible for the output that the user gets.

3. IMPLEMENTATION

There are total ten class labels : airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.
Here is the model implementation.

Image_Classifier

Load training and testing Data from the cifar-10 dataset
(x_train, y_train), (x_test, y_test) = cifar10.load
Convert the data into float32 format
x_train = x_train.astype('float32')/255
x_test = x_test.astype('float32')/255
y_train = utils.to_categorical(y_train)
y_test = utils.to_categorical(y_test)
Prepare the Sequential Model.
Add the Convolutional 2d layer, MaxPooling Layer, Dense Layer, Dropout Layer, and a Flattening Layer.
Compile the complete model together and train it for 30 to 50 epochs at a learning rate of 0.01%.
Select all the necessary units, filters and activation and calculation functions to compute the intended results.

Save the model in an Image_Classifier.h5 format file.

Model_test

Load only the training Data from the cifar-10 dataset
(,), (x_test, y_test) = cifar10.load
Convert the data into float32 format
x_test = x_test.astype('float32')/255
y_test = utils.to_categorical(y_test)
Load the Image_Classifier file
Evaluate the testing results ie, the Test loss, and Test accuracy.
Use the model for a prediction and print all the output for the user to check the accuracy of the model.

IR_Tester

Load the image Classifier model.
Create a user path variable for the user to enter the path of the image file to be categorized as an object.
Load the Image provided in the file path as a 32x32 Image.
Use numpy library to perform numeric conversions and operations on the given Image and finaly test it on the image_classifier module to get the correct output label for that Image.

4. Results

Given in Fig. 1 is the loss and accuracy percentage with the prediction made by the module at the first implementation.

Test Loss = 0.9165961519241334
Test accuracy = 0.6887
Prediction = dog

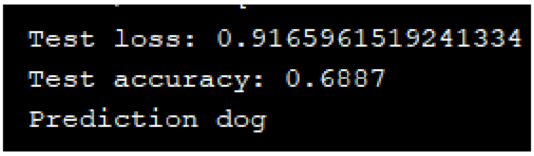

Fig -1: Test Loss and Accuracy

Given in Fig. 3 is the output user might get for the provided input in Fig. 2.

Fig -2: Test Object

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

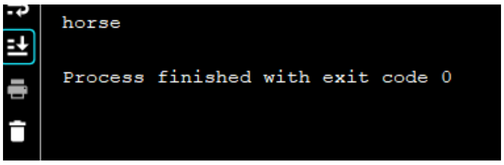

Fig -3: Detection Result

5. Conclusions

We were able to make a object recognizer that can detect a single object in an image based on the concept of Linear regression model. We were able to differentiate between the output that we got for different activation functions and then we were able to choose the most suitable function for our model. We were able to create a module with an accuracy of 67% which can be increased based on the learning rate and the number of epochs.

The model that we created can be used to create an image Classifier file in the form of a pro buff script which can be used for mobile applications and other systems. So we finally created an object recognition model that is fast, space efficient and versatile.

6. References

1. Besl, P.J. and Jain, R.C., 1985. Three-dimensional object recognition. ACM Computing Surveys (CSUR), 17(1), pp.75-145.

2. Logothetis, N.K. and Sheinberg, D.L., 1996. Visual object recognition. Annual review of neuroscience, 19(1), pp.577-621.

3. Felzenszwalb, P.F. and Huttenlocher, D.P., 2005. Pictorial structures for object recognition. International journal of computer vision, 61(1), pp.55-79.

4. Gevers, T. and Smeulders, A.W., 1999. Color-based object recognition. Pattern recognition, 32(3), pp.453-464.

5. Agam, G., 2006. Introduction to programming with OpenCV. Online Document, 27.

6. Lowe, D.G., 1999, September. Object recognition from local scale-invariant features. In Proceedings of the seventh IEEE international conference on computer vision (Vol. 2, pp. 1150-1157). Ieee.

7. Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

8. Cai, Z. and Vasconcelos, N., 2018. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6154-6162).

9. Chen, Y., Li, W., Sakaridis, C., Dai, D. and Van Gool, L., 2018. Domain adaptive faster r-cnn for object detection in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3339-3348).

10. Uijlings, J.R., Van De Sande, K.E., Gevers, T. and Smeulders, A.W., 2013. Selective search for object recognition. International journal of computer vision, 104(2), pp.154-171.

11. Riesenhuber, M. and Poggio, T., 1999. Hierarchical models of object recognition in cortex. Nature neuroscience, 2(11), pp.1019-1025.