OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

OBJECT RECOGNITION USING MACHINE LEARNING

Siddharth Naik [1], Mandar Dike [2,] Sagar Gadgul [3] and Dinesh Satre [4]
Student, Department of Computer Engineering, MMIT, Lohegaon, Pune, India[1,2,3]
Professor, Department of Computer Engineering, MMIT, Lohegaon, Pune, India[4]

Abstract: **We train the system by giving it training examples or as the system gets the training examples itself and eventually adjusting the system's performance until it gives the exact results we want. The system consists of stacked layers of artificial neurons. Each image is fed as input layer, which then talks to the next layer, until finally the output layer is reached. The final results come from the final layer's output.**

The system when given input of multiple images labelled as A, it will try to recognize[7] the labelled image and if the system fails to do so it will learn spontaneously adapting to the new category and crawl over the internet, to train itself to the new input image. Thus, the system uses deep strong neural network[6] which is based on Inception algorithm version 3 which is stacked upon Machine learning program called Tensor Flow. The system will try to identify whether it is already trained on the input, if not, the system will train itself by fetching the data from internet and train the neural network to identify them and go through the input data to give the final output.

Keywords*: Machine Learning, object recognition, object detection, tensor flow.*

---∴∴∴---

I INTRODUCTION

Object recognition is a well-known problem in image recognition. The focus of our system is to combine the power of machine learning with image processing using neural networks[6] to detect objects in the image. The widespread and growing use of online activities and transaction has led to many fraudsters to break down into traditional system. Taking these problems into consideration our focus is to develop a system that can detect and identify objects in an image, so as to fight against the fraudsters. Also, our system can be used in image verification in different process with a wide spread application in online image search, puzzle solving and is expected to grow in coming future. Creating accurate machine learning models capable of localizing and identifying multiple objects in a single image remains a core challenge in computer vision.

The Tensor Flow framework makes it easy to construct, train and deploy object detection models Algorithms like INCEPTION and Tensor Flow can be used to overcome these problems. Paper is organized as follows. Section II describes automatic object detection using Inception algorithm, the operation of system and the terms used in the system. The system architecture diagram represents the working of system. After the input is given, the modification commands and accuracy parameters used are given in Section III. Section IV presents the system architecture. Section V presents experimental results showing results of images tested. Finally, Section VI presents conclusion.

II RELATED WORK

Sometimes an image may contain transparency in it. Detecting these can be very important, and removing these is important in the context of object detection[10][12], and for aesthetic reasons.

Training from the Internet data:

Instead of giving the algorithm our data. The data should be collected by crawling the internet but while doing this there could be a possibility of getting images of different extension, which might be corrupted in some cases. For this purpose, we filter out all the non-JPEG files as we cannot limit only downloading JPEG files. Once all the images are downloaded we filter them accordingly.

We train our system by giving it thousands of training examples and slowly adjusting the parameters until it gives optimal and best results.[4] The network contains 10 to 30 stack layers of artificial neurons. When any image is given as a input. It communicates with the next layer eventually reaching the next layer.

The most challenging part of neural networks was to understand what goes on at each level. After training each layer extract many higher level of features in the image. until the final layer identifies what the image is.[4]

Example: The first focuses on edges and corner the middle layer focuses on basic features and looks for components like a chair or leaf. The final layers combine these into complete interpretations. These neurons activate in response to things like building or tree. We train the network in such a way that they learn the way we want.[1] For inception instead of exactly giving what feature we want to enhance, it also lets the network take its own decision. In this case we give the network an arbitrary image and let the network analyse the image. The network than picks a layer and ask if the network to enhance whatever is detected.

The lower layers tend to produce strokes or patterns because those are sensitive to edges. The higher levels tend to emerge it creates a feedback loop for example if a cloud looks similar to bird the network will make it look like a bird.

1.Machine Learning: Machine learning is branch of computer science that gives a system the ability to learn without being programmed. Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. Machine learning has a strong background of mathematical boost, which results methods, subjective and application domains to different fields. Machine learning is most of the time combined with data mining, which has a greater focus on exploring data which is also known as unsupervised learning. Machine learning can also work without supervision and can be used to establish a basic profile for various entities which can be used to find meaningful deviation.

2.TensorFlow: TensorFlow[3] is a machine learning system that works on a large scale and in different and cross platform environments. TensorFlow implements data flow graphs to show computation, state and operations that modify the state. It maps the nodes of a data flow graph across many machines in a cluster, and within a machine across multiple computational devices, including multicore CPUs, general purpose GPUs, and custom-designed ASICs known as Tensor Processing Units (TPUs).[3] This architecture gives flexibility to the application developer: where as in previous parameter server designs the management of shared state is built into the system, TensorFlow enables developers to experiment with novel optimizations and training algorithms. TensorFlow is Google Brain's second generation system. Version 1.0.0 was released on 11 February 2017. While system runs on single devies, TensorFlow allows the use of multiple CPUs and GPUs. TensorFlow works on 64-bit Linux, macOS, Windows, and mobile computing platforms.

3. Inception: Inception[2] algorithm makes use of deep convolutional networks that plays a main role in image recognition performance in recent years. Inception architecture and algorithm has resulted in great performance at a very little computational cost and proven for classification and detection.

4. Docker: Docker[5] is a technology centered around containers. "Containers include the application and all of its dependencies –but share the kernel with other containers, running as isolated processes in user space on the host operating system. Docker containers work on independent infrastructure: they run on any computer, on any infrastructure, and in any cloud." This is creates an efficient environment that is easy to develop for and run on any machine. That was a more consistent option than installing all of the dependencies. This also allowed easy access of shared resources (such as hardware and file directories) between the native applications and the container.

5. Bottlenecks: A bottleneck is a term that we use for the layer before the final layer that does the classification. Every image is used recursively during training. Calculating the layers for each image takes a very large amount of time by making a cache of the output on the disk makes it easy to be reconsidered. The Bottlenecks are stored in temporary bottle necks directory.

III MODIFICATION COMMANDS

Now that we have a trainer and data. The Inception V3 network can be trained. Following are the modification commands used to retain the network.

```
Python**/tensorflow/**tensorflow/examples/eg/image_retrai
ning/retrain.py \
--bottleneck_dir=/tf_files/bottlenecks \
--how_many_training_steps 500 \
--model_dir=/tf_files/inception \
--output_graph=/tf_files/retrained_graph.pb \
--output_labels=/tf_files/retrained_labels.txt \
```

Training accuracy: It is the number of images that are been used for the current training.

Validation accuracy: Precision of any randomly selected image from the set.

Cross entropy: Function that gives information how well the learning process is going.

The system runs about 4000 steps for every step it takes about 10 images randomly, finds there bottlenecks and gets the right prediction from the final layer and compared with actual labels. After all the training sets are complete, a final test is carried out to test the accuracy. This provides the best result from the trained model.

IV SYSTEM ARCHITECTURE

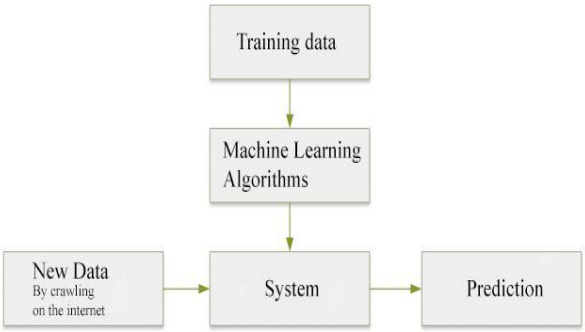

Figure 1: System Architecture

V EXPERIMENTAL RESULTS

Figures shows the results of object detection from an image by using inception algorithm. In this case, we are using an image of an army officer wearing a military uniform. We can see that the system correctly identifies he's wearing a military uniform, with a high score of 0.8.

I tensorflow/examples/label_image/main.cc:206] military uniform (653): 0.834306
I tensorflow/examples/label_image/main.cc:206] mortarboard (668): 0.0218692
I tensorflow/examples/label_image/main.cc:206] academic gown (401): 0.0103579
I tensorflow/examples/label_image/main.cc:206] pickelhaube (716): 0.00800814
I tensorflow/examples/label_image/main.cc:206] bulletproof vest (466): 0.00535088

The system when given input of multiple images labelled as A, it will try to recognize the labelled image[7][8]and if the system fails to do so it will learn spontaneously adapting to the new category and crawl over the internet, to train itself to the new input image. Thus, the system uses deep strong neural network which is based on Inception algorithm version 3 which is stacked upon Machine learning program called Tensor Flow. The system will try to identify whether it is already trained on the input, if not, the system will train itself by fetching the data from internet and train the neural network to identify them and go through the input data to give the final output as recognised objects[11][13] in the image.

Figure 2. Various objects detection on the basis of accuracy score:

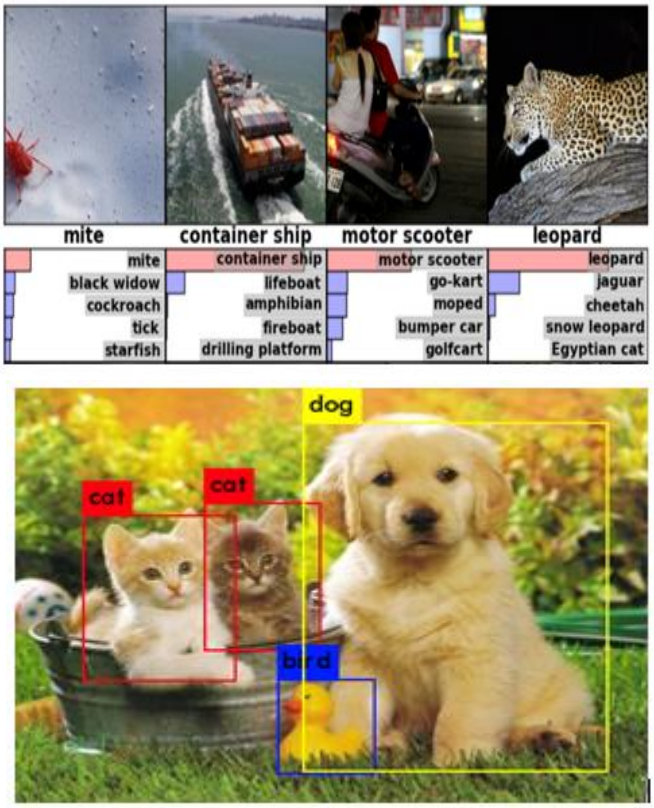

Figure 3 Object detection in an image with multiple objects in the same image.

VI CONCLUSION

We have implemented an automatic object detection technique from an image. Our algorithm successfully detects the objects from the image which consists of multiple objects as well. We have applied our algorithm on many images and found that it successfully detects the objects in the images.

REFERENCES

[1] Large Scale Visual Recognition Challenge 2016 (ILSVRC2016). (n.d.). Retrieved December 03, 2016, from http://image-net.org/challenges/LSVRC/2016/results

[2] (2015). Inceptionism: Going Deeper into Neural Networks. Retrieved December 03, 2016, from https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

[3] TensorFlow is an Open Source Software Library for Machine Intelligence. (n.d.). Retrieved December 03, 2016, from https://www.tensorflow.org/

[4] (2015, December 07). How to Classify Images with TensorFlow. Retrieved December 03, 2016, from https://research.googleblog.com/2015/12/how-to-classify-images-with-tensorflow.html

[5] (n.d.). Retrieved December 3, 2016, from https://www.docker.com/what-docker#/VM

[6] (2015)SupervisedimageclassificationusingDeepConvolutionalWaveletsNetwork
[7] (2016)IAmRobot:(Deep)LearningtoBreakSemanticImage CAPTCHAs
[8] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using hard ai problems for security," in EUROCRYPT '03.
[9] M. Foley, ""Prove You're Human": Fetishizing material embodiment and immaterial labor in information networks," Critical Studies in Media Communication, vol. 31, no. 5, pp. 365–379, 2014.
[10] E. Jeremy, J. R. Douceur, J. Howell, and J. Sault, "Asirra: a CAPTCHA that exploits interest-aligned manual image categorization," in CCS '07.
[11] Distil Networks. CAPTCHAs Have Negative Impact on Web Traffic and Leads. http://www.distilnetworks.com/distil-networks-study-captchas-negative-impact-on-web-traffic/.
[12] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage, "Re: CAPTCHAs: understanding captcha-solving services in an economic context," in USENIX Security '10.
[13] E. Bursztein, J. Aigrain, A. Moscicki, and J. C. Mitchell, "The end is nigh: Generic solving of text-based CAPTCHAs." in USENIX WOOT '14.