

The-Phy Lieng

Tensorflow Facial Recognition – jsTensorflow - OpenFace

ABSTRACTION

According to Andrew Ng, formerly head of Google Brain, machine learning has a ton of day-to-day life applications ranging from self-driving cars and practical speech to effective web search and face recognition. To investigate how machine learning works, We use Tensorflow as a mean and deploy the structure of Tensorflow itself. We use the original research Tensorflow models with two models configuration: Faster-RCNN and SSD-MobileNet for face recognition. The dataset is extracted from 8 members of Android Department to obtain 80 samples for the first expereriment and 498 samples for 2 members for the second experiment using. Although the results of the two process are completely failed due to lack of data and the data standard respectively, We find out a more simple yet efficient method is to use the OpenFace and dlib python library which specializes in facial recognition system. The dataset used in Tensorflow project is then made use of as a resource for two different experiemnent one with the original dataset and one with difference in image's contrast. The experiment shows that with the orignal dataset, OpenFace cannot recognizes any faces in the test pictures but the adjusted intensity using sauna filter effect dataset give out 100% accuracy. Besides, the jsTensorflow project is made along with the time of training models with the result of a game using socket to communicate among the server and game client.

Index Term,, Tensorflow, face recognition, models,

I. INTRODUCTION

Tensorflow is an open source machine framework which originally developed by Google Brain team within Google's AI organization [1]. There are a variety of projects in the Tensorflow ecosystems such as Tensorflow Lite, Swift for Tensorflow, Magenta. However, this report is only talking about Tensorflow Models and Tensorflow.js.

Face recognition and detection has become a part of humans everyday life which is a special feature of several reknown social application. According to Jared Bennett, Center for Public Integrity, there are 2 billion monthly users who upload 350 million photos every day on Facebook which provides the company an infinite sources of face dataset[3].

There has been a rise in objection detection and recognition, eg face recognition until the year of 2018. The Taylor Swift detector [4] is an example. It also used the Tensorflow Object Detection API[1], with MobileNet pre-trained model as a checkpoint for transfer learning. The problem of this project is it used the dataset orignally from Google Images with high resolution which creates a barrier of taking dataset not from Google Images and what is the quality of the image can be used to train. //Therefore, there should be some methods to figure out how suitable the dataset to reduce the time and cost wastefor training?

This research tries several different strategies to have a bigger overview of how each methods affect the result of face recognition. The main contribution of our report is that the

quality and quantity of the dataset should be considered carefully before using for Tensorflow models. Furthermore, in term of face recognition and detection, we found out a more reliable yet simple and fast method is to use OpenFace[5] and dlib[6] open source library. In the end, the purpose of this research is to figure out how Tensorflow and OpenFace work on face recognition and their applications.

In the first experiment with Tensorflow, the authors collect all the dataset by using Samsung Galaxy A6 with camera 16mp. Next, the authors preprocess images by resizing, labeling, splitting them into training and test sets, and then convert those images into test and train record file for feeding into Object Detection API. In the second experiment with Tensorflow, the collecting dataset step is the same with the first one. However, now there are only two classes with 498 samples in total. Both the above experiments using MobileNet on NVIDIA Quadro K2000 to train the model. The result of all above experiments all completely failed after nearly 200,000 training steps.

In the first experiment with OpenFace, the authors use the original dataset from the experiment with Tensorflow. Using the k-nearest-neighbors (KNN) algorithm for face recognition, the result failed again with the error that the number of neighbors are bigger than the number samples needed. However, after modifying the image intensity, the accuracy when using OpenFace can reach to 100% with adjusted distance_threshold equals 0.4.

The rest of this report is structured in the following way. In section II, we present about the applications and some related works of Tensorflow and OpenFace. The important procedures of our Face Recognition works are demonstrated in section III using Tensorflow Object Detection API [6] and OpenFace [5]. Section IV displayed experiment results and the analysis. The conclusion and related works are presented in section V.

II. Application and Related Works

A. Tensorflow

There are a lot of Tensorflow projects which are applied in a variety of fields. A JavaScript library, Tensorflow.js [7], which is implemented for training and deploying ML models in the browser and on Node.js. One of the Tensorflow.js project is Teachable Machine[8] which uses deeplearn.js library for everyone to explore machine learning. The experiment can be used live in browser without any coding required.

Furthermore, Tensorflow also provides several pre-trained classifiers as well as models which anyone can use to train their own convolutional neural network objection detection from scratch. Edge Electronics who uses Tensorflow Object Detection Classifier to train the Pinochle Deck [9], create a playing card detector that can accurately detect nines, tens, jacks, queens, kings, and aces. The tutorial instructs the step for setting up and installation on Windows 10 which the authors also use as a tutorial.

B. OpenFace

Florian Schroff and et al. proposed a unified embedding for FaceRecognition and Clustering which also known as FaceNet[11]. OpenFace is a Python and Torch implementation of face recognition with deep neural networks based on the above paper. Face Recognition library Matt Kiser et al. at Algorithmia [12], created automated front desk A.I using CMU OpenFace library that uses facial recognition to detect and greet the workers when they arrive at the office. The Face Recognition library based on OpenFace which is made by Adam Geitgey [13]. It is built using dlib's state-of-the-art face recognition built with deep learning which has a remarkable accuracy of 99.38% on the Labeled Faces in the Wild [14]. The authors also use this library as a mean to implement face recognition.

III. METHOD

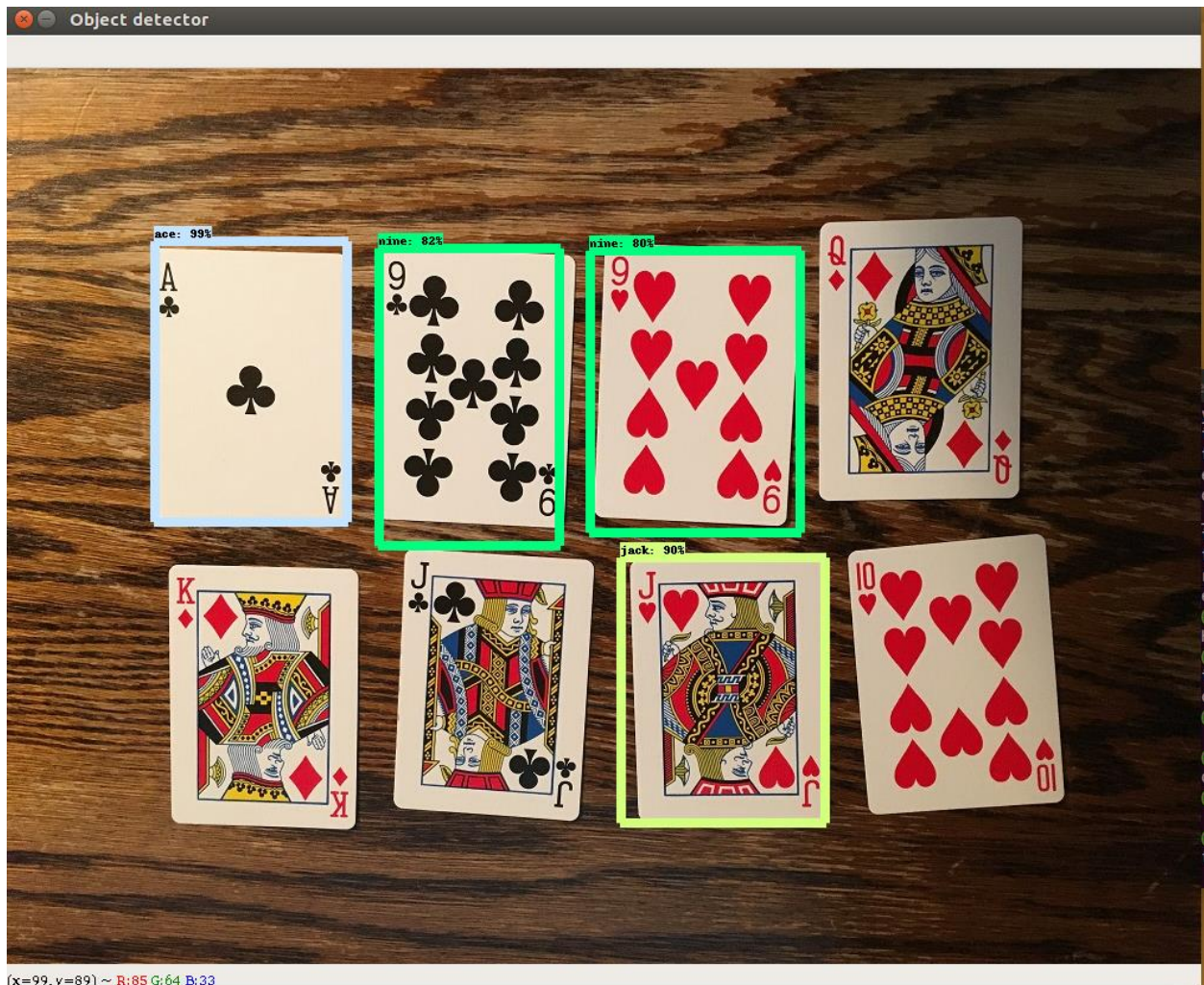
A. Tensorflow Object Detection API

1) RCNN – faster rcn

a. SSD MOBILE NET.

B. Face Recognition library.





EXPERIMENT AND RESULT

CONCLUSION

- [1] <https://www.tensorflow.org/>
- [2] <https://github.com/tensorflow/models>
- [3] <https://www.thedailybeast.com/how-facebook-fights-to-stop-laws-on-facial-recognition>
- [4] <https://towardsdatascience.com/build-a-taylor-swift-detector-with-the-tensorflow-object-detection-api-ml-engine-and-swift-82707f5b4a56>
- [5] <https://github.com/cmusatyalab/openface>
- [6] <https://dlib.net>
- [7] <https://github.com/tensorflow/models/tree/master/official>
- [8] <https://js.tensorflow.org>

- [9] <https://teachablemachine.withgoogle.com/>
- [10] <https://github.com/EdgeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10>
- [11] <https://cmusatyalab.github.io/openface/>
- [12] <http://blog.algorithmia.com/2016/02/hey-zuck-we-built-your-facial-recognition-ai/>
- [13] https://github.com/ageitgey/face_recognition#face-recognition
- [14] <http://vis-www.cs.umass.edu/lfw/>