TEST Engineering & Management

January - February 2020
ISSN: 0193-4120 Page No. 16736 - 16742

All in One Computer Vision Application using MLkit

Priya Batta[1], Sugandhi Midha[2],Robin Sinha[3], Manish Kumar[4]

[3,4] Student

[1,2,3,4] Department of Computer Science, Chandigarh University Mohali, Punjab

Article Info
Volume 82
Page Number: 16736 - 16742
Publication Issue:
January - February 2020

Article History
Article Received*: 18 May 2019*
Revised*: 14 July 2019*
Accepted*: 22 December 2019*
Publication*: 29 February 2020*

Abstract

Methods of deep learning are broadly used in smart systems which include powerful computers, but now a days they are also used in mobile systems. To perform pre-trained prototypes on mobile system frameworks, application programming interfaces have been released. In this research, various modules have been developed using the Mlkit firebase framework of Google using Flutter Language.

I. INTRODUCTION

Deep Learning is widely known technique now a days in a machine learning and has become very popular in many different applications that include classification, voice recognition, medical fields etc. Convolution neural network is well liked deep learning algorithm used to detect objects with its classification muddles. Neural convolution network provides both extraction and classification functions as opposed to other machine learning algorithms. While deeper architectures deliver higher results, we need a vast array of knowledge about training and powerful computers. The training time for a model can take up to several hours. Several applications use pre-trained models such as Google Net[1], OverFeat[2], etc. While deep learning is getting popular in many ways, there are various systems of testing, training are added day by day. TensorFlow[3], Caffe[4], Keras[5] are few examples of architecture of systems which were developed for processing deep learning models.

Nevertheless, these systems have been developed for computers with specific features. In addition to improving the approaches to machine learning, mobile systems have become a vital part of our living standards. In fact, there are machine learning applications that run on mobile systems. These widespread and powerful fields are brought along by some tools. MLKit[6] and TensorFlow are the two programming interfaces (APIs) that enable users to use deep learning models in mobile systems. MLKit development began at Edinburgh University in 1989. The project's main objective was to incorporate the advantages of victimization in an application-oriented language with a template that stores data, however much space and time required. It's four different versions. It's a trash collection feature in the latest version, a bytecode backend that offers system flexibility. Another property of MLKit is to allow you to use device and cloud machine learning skills. For base the mobile development system, it is combined. MLKit includes identification of text, face and landmarks, scanning of barcodes, marking of objects. In contrast, MLKit helps computer programmers to migrate their own non-fat TensorFlow designs. TensorFlow is an associate in the deep learning system of the open source platform Nursing founded by Google Brain on November 9, 2015. Google was victimizing the model established by DistBelief[7] in 2011. TensorFlow has been designed to be more easily

Published by: The Mattingley Publishing Co., Inc.

January - February 2020
ISSN: 0193-4120 Page No. 16736 - 16742

suited for Google's new by-products and research. TensorFlow is more useful than DistBelief, faster, smarter, and versatile. TensorFlow is therefore referred to as the second-generation method. Multiple CPUs and GPUs will run TensorFlow. This is also used on entirely different networks. Across 64-bit UNIX systems, for example, macOS, Linux, and mobile computing. Google explicitly uses the TensorProcess Processor (TPU) for TensorFlow and machine learning. TensorFlow supports identification, speech, sound and object recognition, and works on mobile systems for text applications.

II. RELATED WORK

Bremananth[8] suggests, among the different methods of coin identification, a architecture that focuses on numerals instead of front or back pictures. The aim is to take a coin image and retrieve the number using a pattern matching technique. The methodology uses many strategies to reliably identify these numbers, such as numerical colour, threshold form, Gabor sorting, and back propagation networks.

Kim[9] wants to use CNN to correctly recognize Imperial Roman coins. The research applies a hierarchical structure using CNN structures for activities related to the sorting of coins. The goal is to find a symbol on the photographs of the coin. As the choice of a set of coin elements that represents the group, an optimization problem is devised. The parts selected are considered as elements that can be used to discriminate against the coin. Such icons can be important to the numismatic experts ' study of coin characteristics.

Modi[10] created an artificial NN focused on Indian coins being automatically recognized. Both sides of coins can be recognized by the system. Information from images is extracted using methods like Hough Transformation, Pattern Averaging, etc. The information extracted is used as input for the training of an NN.

Computation offloading processes were thoroughly studied[11],[12],[13],[14] and were typically classified into two categories: coarse grain offloading[11],[12] and fine grain offloading[13],[14]. Coarse-grained discharge also applies to absolute discharge or full discharge in which the entire process is transferred to the internet. This method does not require an overhead estimation pre-execution estimate. Therefore, fine-grained offloading (partial offloading or adaptive offloading) automatically transmits as little code as necessary and only downloads the computer-hungry parts of the software.

Alvaro Arcos-Garcıa, Mario Soilan, Juan A. Alvarez-Garcıa, Belen Riveiro,[12] discussed in paper that Using synergies between remote navigation sensors system and deep learning models for traffic sign recognition systems, Expert Systems Technologies can be used in Traffic Sign Recognition Systems (TSRS) which will be very useful for Advanced Driver Assistance Systems (ADAS) and making autonomous cars, but with a wide range of problems that need to be tackled with new technologies. Moreover, there is also a very wide and varied variety of different traffic signs when it comes to differentiate between different countries.

For example, in Spain (Spain, 2003), Germany and Belgium there are more than 200 traffic sign categories. All of these problems concern with TSRS systems and are very important factors which must be taken into account when dealt with.

The major difference between these ways and the work we are discussing in this study is that according to membership class, we find and distribute the whole coin. During the training phase, the data will be collected from the coin directly and transmitted to Neural Network. Therefore, we obtain a classification system that can foresee such coin categorization.

Published by: The Mattingley Publishing Co., Inc.

January - February 2020
ISSN: 0193-4120 Page No. 16736 - 16742

The architecture of the application contains different modules fitted in one application.

The background of application works on simple method with is done by the Google's Firebase MLkit which can be changed into data flow diagram.

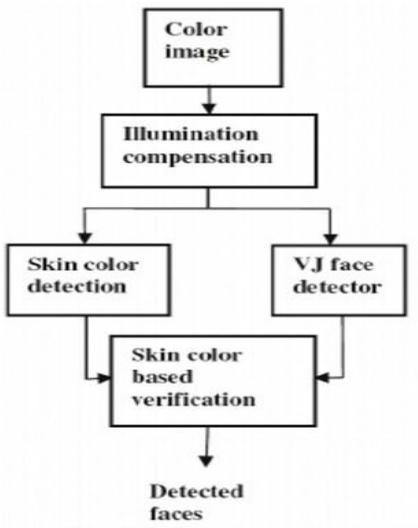

Fig 2. Data Flow Diagram for Image Processing

III. PROPOSED WORK

Traditionally, the computer vision was used in various fields but it required powerful processors which were only available in computers. With the advancement of mobile computing and coming of mobile with strong processors there is a hope that we can do processes like face recognitions, OCR image processing using our mobile phones. With packages like MLkit we can easily process image with our normal mobile phones. With use of android studio and google flutter and integrating it with MLkit, we can easily develop an application which can perform all the task in just one application.

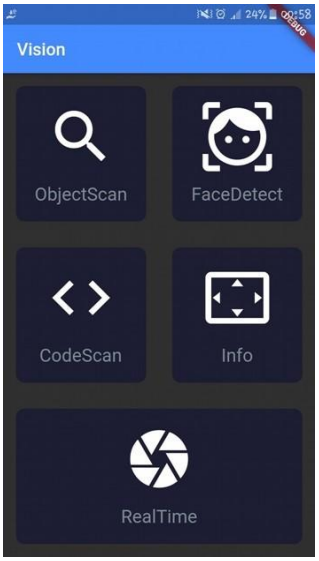

Fig 1. Our application vision with different modules

This application contains different modules. Each module has different functions. We are calling it "All in One Computer Vision.

The Novelty factor in our approach is we are providing multiple features of computer vision in just one application. A application which will act as a single solution stop for all type of Computer Vision Problems such as Face recognitions, Object Detections, Code Scan, Text Scan in a single click.

So our first module in Fig 3 is Text Scan:

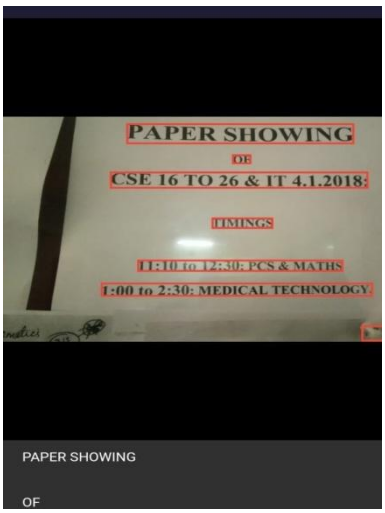

Fig 3. Text Scan Module Displaying Working Image

Published by: The Mattingley Publishing Co., Inc.

January - February 2020
ISSN: 0193-4120 Page No. 16736 - 16742

This module scans every single text present on the image and shows it to users. It helps user to use the text indepentdently from the image. In Fig 3. We have detected the text from the image capture from the mobile phone. The test shows that each

The second module in Fig 4 contains Code Scan:

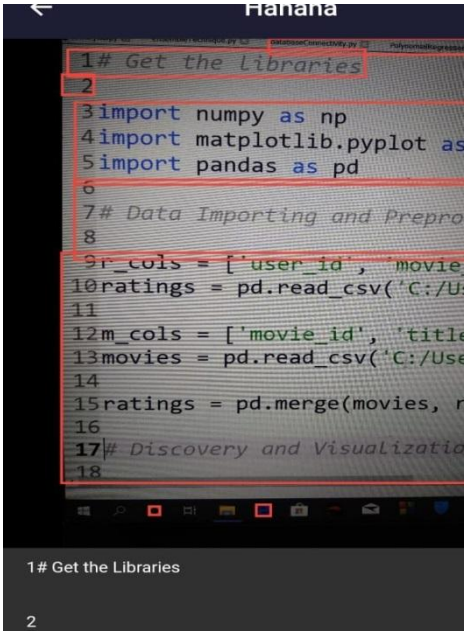

Fig 4. Code Scan Working Module

The code scan module is generally developed for programmers so that they easily capture and picture of the code from any image and easily use code scan to take out all the codes from the image. For example it can be very easy to scan the code from hardcopy.

The third module in Fig 5 is Face Detection:

The face detection depends on various parameters. Face detection can be considered as an object-class detection case in particular. The task is to find locations and sizes of all objects in an image belonging to a particular class when detecting objects. E.g. top torsos, foot-drivers, and cars.

The algorithms of face-detection focus on the detection of human faces on the front. This is analogous to the object recognition where a person's image is bit by bit matched. Picture matches to the database's image stores. Any changes in the database facial features will invalidate the corresponding process.

Similarly, in Fig 3 text generation works as to generate the texts outputs from the image processing. Texts from the images can be first out lined and returned as the output of the Text recognition module.

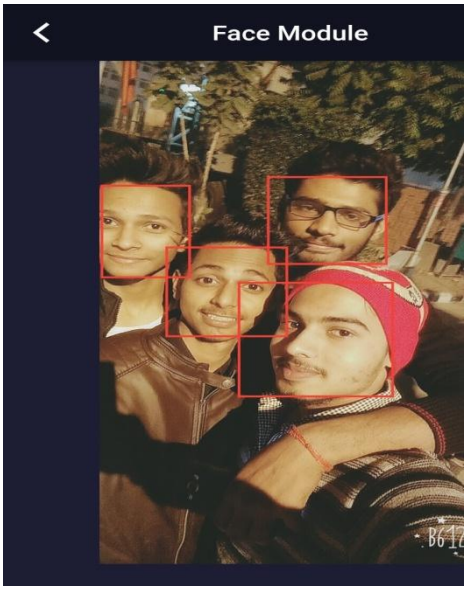

Fig 5. Face Detection Working Module

This module can detect faces from anygiven image. The application also has feature to detect face on run time or we can say real time.

The fourth in Fig 6 module is Object Detection:

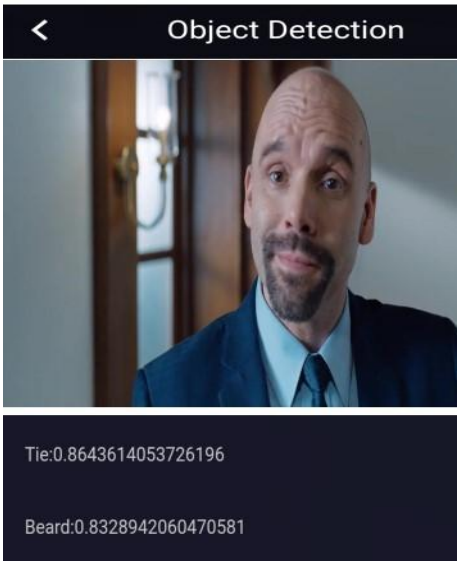

Fig 6. Object Detection Working Module

Published by: The Mattingley Publishing Co., Inc.

TEST Engineering & Management

January - February 2020
ISSN: 0193-4120 Page No. 16736 - 16742

This module can detect various objects from the images in real time as well as from pictures. The module works on prediction and also specifies the percentage of surety. For there is beard in the image and 77 percent sure that there is a moustache present in the image.

The fifth module in Fig 7 is QR scanner:

Fig 7. QR scanner module

This module scans the given QR code and returns the output. There are many other modules such as Barcode Scanner, Label etc.

All this modules are a part of single application Vision.

III. RESULTS

We have done two things. We have worked on face module and Code scan module. Which both are working on Google's Firebase MLkitFramework.

	MLKit
Face Detection	✓
Text Detection	✓
Barcode Detection	✓
Label Detection	✓
Classification	✓

Table I. The Mlkit API workings

In Table I. the results are shown when MLkit API was tested on various functions like Face, Text, The MLkit was able to detect Faces, Texts, Barcodes, Labels, Classifications. The MLkit was unable to recognize the speech otherwise we would have implement speech unlock in the application.

	MLKit
Accuracy %	50
Ease of installation	Firebase required
Resource usage	Less

Table II. The MLkit requirements and Performances

The MLkit performs very with less resource usage. With optimization techniques we were able to brig accuracy above 80% in our application.

IV.CONCLUSIONS

We believe that this first step in understanding how to use deep learning in mobile environments provides a basis for more comprehensive studies and will contribute to important innovative designs for sensing applications being created. In the widespread adoption of consumer-ready sensing technologies powered by deep learning, our research touches only the technical surface of a breakthrough. Deep Learning is widely known technique now a days in a machine learning and has become very popular in many different applications that include classification, voice recognition, medical fields etc. CNN or Convolution neural network is a simple but well-known learning algorithm used in recognizing of various classification problems in modern world. In different fields of use, such as medical diagnosis, text creation, recognition, etc, deep learning techniques are widely used and return best results than other methods. While these models are trained on strong machines, pretrained models on various platforms such as mobile systems are used. There is a good a scope of this application as now a days in

January - February 2020
ISSN: 0193-4120 Page No. 16736 - 16742

every single thing, Machine Learning is being used so with advancements of technology this application will grow at the next level. With use of Dart in google flutter, the development of the application is very easy while working with deep learning model API's like MLkit. The MLkit API is good and has various usages. It can be developed even more to increase its accuracy. We noticed that MLkit was working poor than TensorFlow but TensorFlow used a lot more resources than MLkit. So, we decided to use Mlkit and not TensorFLow. We also noticed that MLkit was not able detect object accurately as it was confused between similar things like jacket and suit. This issue can be solved with development of the API then only these issues can be solved and then our application performance will go higher. We have developed the application to merge the different things we can do in computer vision project so that all the work can be done in a single application. This is one solution to many different projects combined in one single project or more specifically an android application, which can be treated as a single stop to all the computer vision related projects like object detection, face recognitions etc.

REFERENCES

[1] C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, and A Rabinovich, "Going deeper with convolutions", CoRR, abs/1409.4842, 2014.

[2] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks", arXiv:1312.6229v4, 2014.

[3] Tensorflow, https://www.tensorflow.org, Accessed: 24.06.2018.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long,
R. Girshick, S. Guadarrama, T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding", arXiv preprint arXiv:1408.5093,

[5] Keras: The Python Deep Learning library, https://keras.io, Accessed : 24.06.2018.

[6]B. Elbouchikhi, "Introducing ML Kit", https://developers.googleblog.com/2018/05/introduc in g-ml- kit.hml, May, 2018.

[7] J. Dean, G.S. Corrado, R. Monga, K. Chen, M. Devin,Q.V. Le, M.Z. Mao, M.A. Ranzato, A. Senior, P. Tucker, K. Yang, A. Y. Ng., "Large Scale Distributed Deep Networks", NIPS, 2012.

[8] R. Bremananth, B. Balaji, M. Sankari, and A. Chitra, "A New Approach to Coin Recognition using Neural Pattern Analysis," in 2005 Annual IEEE India Conference - Indicon, pp. 366–370, Dec 2005.

[9] J. Kim and V. Pavlovic, "Discovering Characteristic Landmarks on Ancient Coins using Convolutional Networks," CoRR, vol. abs/1506.09174, 2015.

[10] S. Modi and D. S. Bawa, "Article: Automated Coin Recognition System using ANN," International Journal of Computer Applications, vol. 26, pp. 13–18, July 2011. Full text available.

[11] H. Eom, P. S. Juste, R. J. O. Figueiredo, O. Tickoo, R. Illikkal, and R. Iyer, "Machine learning-based runtime scheduler for mobile offloading framework," in IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC 2013, Dresden, Germany, December 9-12, 2013, 2013, pp. 17–25.

[12] H. Eom, R. J. O. Figueiredo, H. Cai, Y. Zhang, and G. Huang, "MALMOS: machine learning-based mobile offloading scheduler with online training," in 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2015, San Francisco, CA, USA, March 30 - April 3, 2015, 2015, pp. 51–60.

[13] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," IEEE Trans. Mobile Comput., vol. 14, no. 12, pp. 2516– 2529, Feb. 2015.

[14] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile

Published by: The Mattingley Publishing Co., Inc.

January - February 2020
ISSN: 0193-4120 Page No. 16736 - 16742

computing," IEEE Trans. Wireless Commun., vol. 11, no. 6, pp. 1991–1995, Jun. 2012.
[15] Alvaro Arcos-Garc ´ ´ıa, Mario Soilan, Juan A. ´ Alvarez-Garc ´ ´ıa, Belen Riveiro, ´ Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems, Expert Systems With Applications (2017)

Published by: The Mattingley Publishing Co., Inc.