PLEASE TURN OVER (PTO), MUSIC NOTATION PAGE TURNER USING CONVOLUTIONAL NEURAL NETWORK

D.T.D.M. Dahanayaka*, R.J. Wimalasiri

Department of Mechanical Engineering, The Open University of Sri Lanka

INTRODUCTION

Listening to music is one of the most enjoyable human experiences. The importance of music has increased in our stressful daily lives. Music notation or musical notation is a set of symbols used in written music. It allows musicians to know which note to play for how long. Usually, paper is the medium of the music sheet, though it does not have much flexibility. This study proposes an image sliding/turning Android mobile application 'PTO' instead of paper music sheets. There are various types of image sliding applications (apps) but they are not popular among musicians since they are not user friendly. In certain applications, sliding of the images has been done by touching the screen which is difficult while performing. Some of them automatically slide images at pre-determined intervals which could also vary under different audiences and performing techniques. Once the app is developed, a powerful machine/cloud with graphics processing units (GPUs) needs to be used to train the model. In this study, 'Google Colab' was used to train the model. Google Collaborative is a powerful platform for developing machine learning models in Python. It is based on the Jupiter notebook and offers free access to GPU. To train the images, the Convolutional Neural Network (CNN) was used. It extracts the features of the image and converts it into a lower dimension without losing its characteristics [1, 2, 3, 4]. There are various advanced algorithms for CNN such as You Only Look Once (YOLO), Single Shot Detector (SSD), and Faster Region CNN (F-RCNN) [5, 6, 7, 8]. In this study, SSD algorithm was used and the functionality of the SSD is shown in Fig.1. In this project, SSD300 was used to train the images. SSD uses VGG16 (Visual Geometry Group) to extract feature maps. SSD adds six more auxiliary convolutional layers after the VGG16. The VGG16 was used because of its strong performance in high-quality image classification tasks.

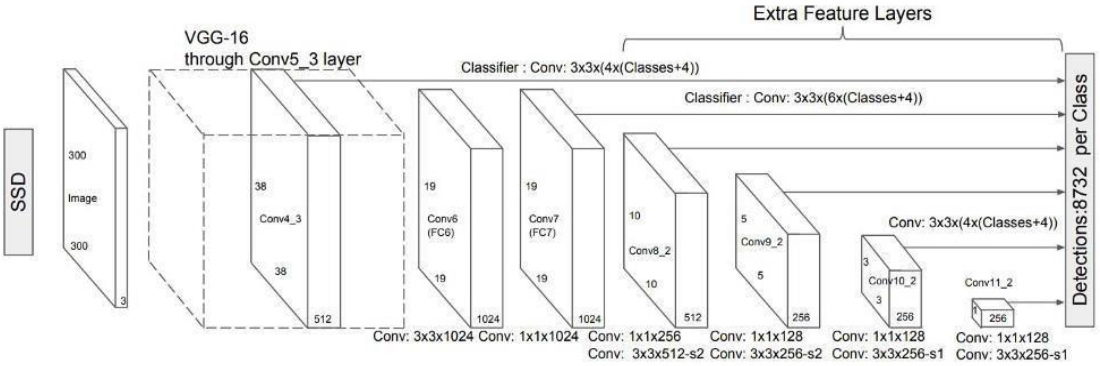

Fig.1. The architecture of a Single Shot MultiBox Detector [6]

'Android Studio' and 'TensorFlow Lite' were used for Android application development [9, 10]. TensorFlow Lite is more appropriate due to the limited capacity and processing power of a mobile phone. First, it converts the trained TensorFlow model to the TensorFlow Lite file format (.tflite) using the TensorFlow Lite converter and this TensorFlow Lite supports Android and iOS platforms.

METHODOLOGY

Image training process

The block diagram of the process using which the images were trained is shown in Fig.2. First, the images were divided into classes and then annotation (process of labelling the available data) was done for all the images as labelled data sets are necessary for supervised machine learning. Then, the images were trained using CNN and the trained classifier was obtained.

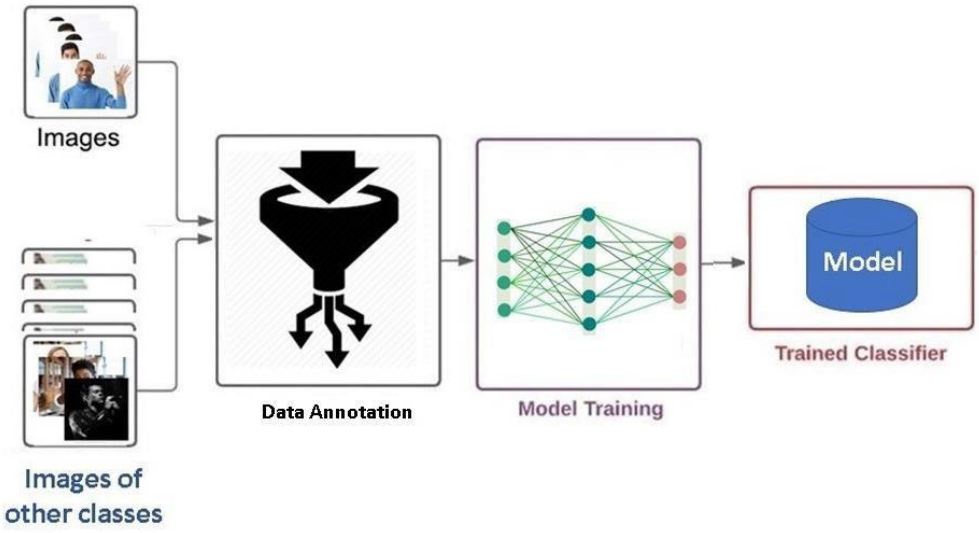

Fig.2. The block diagram for the image training

Two classes of images were used for this study. They are "hand" and "other". 2000 images were used for each class. All the images were converted to 300×300 fixed pixel size. As shown in Fig.3, the annotation was then conducted for all the images. The images were split into training data (80%) and testing data (20%).

Fig.3. Annotation of the image

The Google Colab and SSD algorithm were used for the training process. Then, the trained model was converted into '.tflite' using TensorFlow Lite Converter.

Android Mobile Application

Android Studio was used to develop the Android mobile application. 'Java' language and 'compileSdk-Version 28' were used to develop the new Android Studio application. Then, the user interface layouts were created (Fig.4).

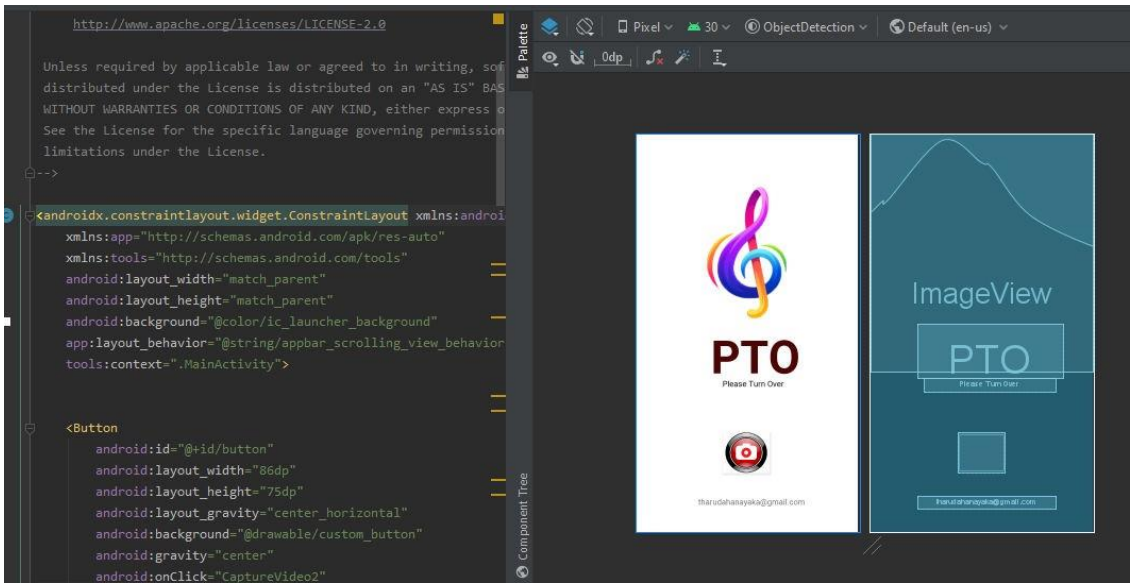

Fig.4. The user interface layout of the app

Image Prediction

The block diagram for image prediction by the app is shown in Fig.5. The trained classifier (model) was used to recognize the hand gestures. Using the selfie (front) camera the real time video is captured. Then, the video is converted to frames. After that, the trained model was used for prediction so that image sliding/turning can be accomplished without touching the screen.

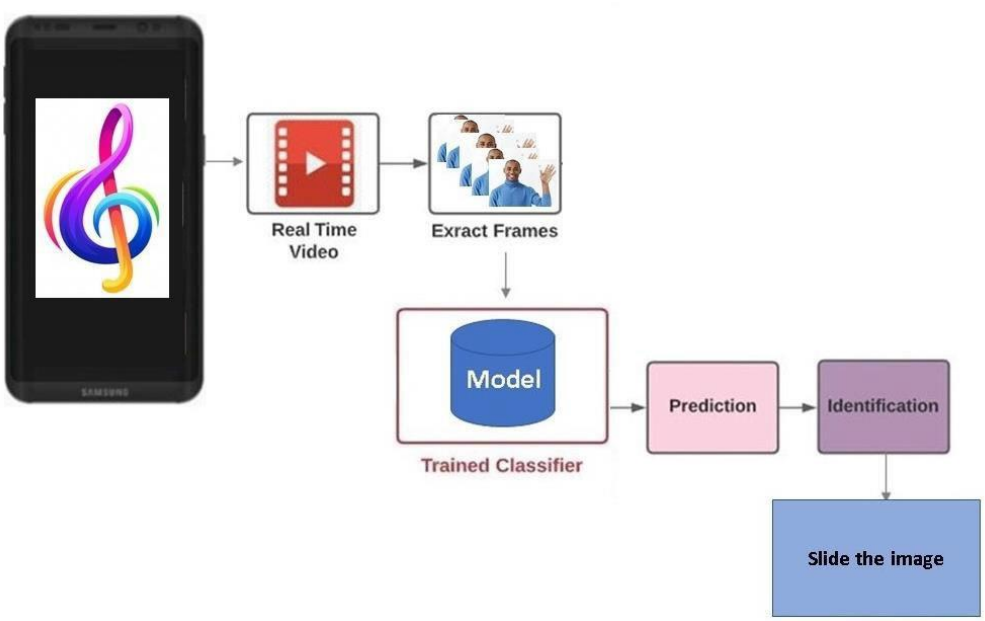

Fig.5. Block diagram for image prediction by the mobile application

RESULTS AND DISCUSSION

Image training results

The progress of the training process of the model can be monitored using the TensorBoard graph. These graphs provide visualizations for the machine learning workflow. The most important graph is the Loss graph since it illustrates the model loss. If the model makes fewer errors, that means the model has learnt all the details. The Loss Function (L) is a combination of Classification (Confidence Scores, L_{conf}) loss and Regression (Localization, L_{loc}) loss.
Loss Function is,

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))$$

The Regression loss is the mismatch between the Ground truth bounding box and the Predicted bounding box. The equation for the Regression loss is,

$$L_{loc}(x,l,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} smooth_{L1}(l_i^m - \hat{g}_j^m)$$

The Classification loss means the loss of making a class prediction. The equation for the Classification loss is,

$$L_{conf}(x,c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0)$$

where,

$$\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

N : number of matched deault boxes α : scaling factor for localization loss

l : predicted bounding box g : ground truth bounding box

$\hat{g}$: encoded ground truth bounding box

$\hat{c}_i^p$: softmax activated class score for default box i with category p

x_{ij}^p : matching indicator between deault box i and Ground truth box j of category p

x_{ij}^k : matching indicator between deault box i and Ground truth box j of category k

The Classification loss graph and Regression loss graph are shown in Fig.6 and Fig.7. This graph shows that the loss is decreasing. It suggests that the model has found a solution. The Total Loss graph is shown in Fig.8.

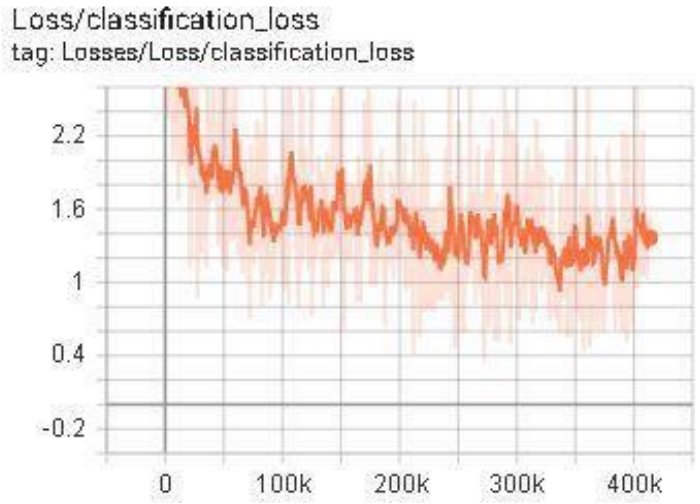

Fig.6. Classification loss graph

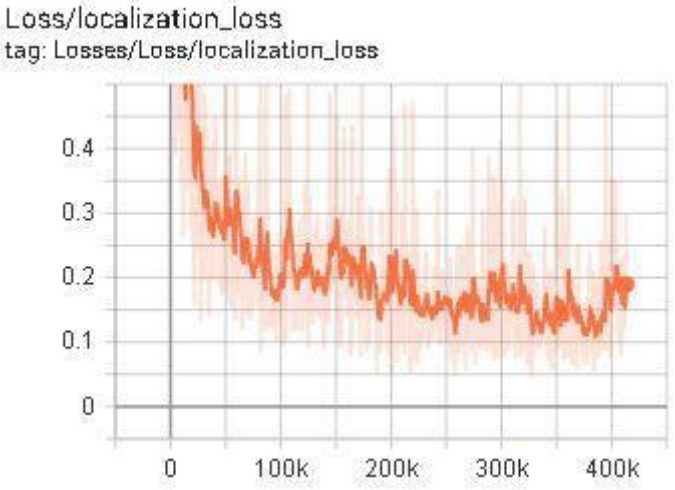

Fig.7. Regression loss graph

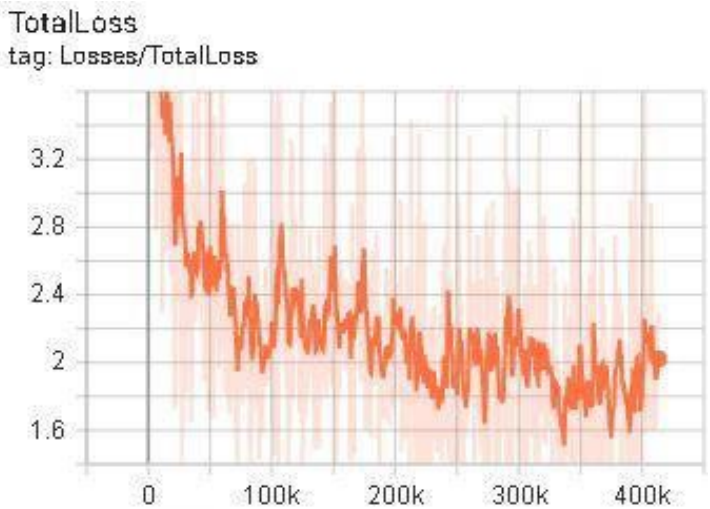

Fig.8. Total loss graph

Mobile Application Testing Results

First, the real-time video is converted into images and then the recognition is done. After recognizing the specific hand gesture, the image can slide/turn. The user interface layout of the PTO Android mobile app is shown in Fig. 9 and the identification of the specific hand gesture is shown in Fig. 10.

Fig.9. PTO interface

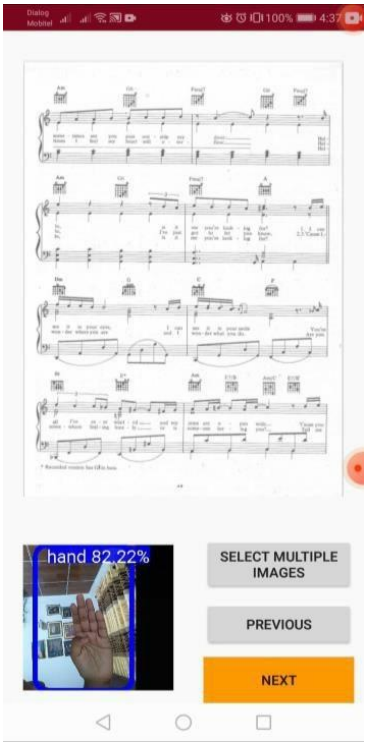

Fig.10. Testing by hand gesture

CONCLUSIONS

While playing instruments it is very difficult to turn pages. There are different types of image sliding apps, but none of them are used regularly by musicians. Hence, this study offers an image sliding/turning Android based application named 'PTO' instead of paper music sheets. GPU is necessary to train many images. Therefore, Google Colab was used because it offers free access to a computer with an appropriate GPU. This study presents an offline mobile app with a user-friendly interface. The proposed app could be used not only when playing musical instruments but also for any situation in which turning of pages is required.

REFERENCES

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *In:arXiv preprint arXiv:1712.04621G*

Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: when to warp? *CoRR*, abs/1609.08764.

Sri. Yugandhar, M., Jayaram, K., Kowshik K., Kanithi S.K., & Jagilinki J. (2020). Handwritten Text Recognition using Deep Learning with TensorFlow. *International Journal of Engineering Research and*, V9(05).Jkjh

Zhao, Z., Zheng, P., Xu, S., & Wu, X. (2019). Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems,* 30(11), pp.3212- 3232.

Kumar, A., & Srivastava, S. (2020). Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector. *Procedia Computer Science,* 171, pp.2610-2617.

A Review on Real Time Object Detection using Single Shot Multibox Detector. (2019). *Journal of Environmental Science, Computer Science and Engineering & Technology*.

Uzun, D., & Bilban, M. (2019). Autonomous Vehicle and Augmented Reality Usage. *International Journal of Engineering and Management Research,* 09(06), pp.39- 43.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *In: arXiv:1409.1556v6y.*

Uzun, D., & Bilban, M. (2019). Autonomous Vehicle and Augmented Reality Usage. *International Journal of Engineering and Management Research,* 09(06), pp.39-43.

Choe, D., Choi, E., & Kim, D. (2020). The Real-Time Mobile Application for Classifying of Endangered Parrot Species Using the CNN Models Based on Transfer Learning. *Mobile Information Systems*, 2020, pp.1-13.