American Journal of Advanced Research, 2019, 3–1
July. 2019, pages 6-12
doi: 10.5281/zenodo.3264023
http://www.ibii-us.org/Journals/AJAR/
ISBN N 2572-9829 (Online), 2572-9810 (Print)

Skin Lesion Detection by Android Camera based on SSD- Mobilenet and TensorFlow Object Detection API

Arwa Mohammed Taqi [1] *, Fadwa Al-Azzo [1], Ahmed Awad [1], and Mariofanna Milanova [2]

[1]Department of System Engineering, [2]Department of Computer Science, University of Arkansas at Little Rock, Arkansas, USA

*Email: ahmohammedt@ualr.edu, arwa.alattar@uokerbala.edu.iq

Received on April 16, 2019; revised on June 20, 2019; published on June 30, 2019.

Abstract

With the fast evolution of medical imaging study, a great interest in skin cancer detection has been investigated with numerous computer algorithms. Generally, skin lesions are examined with a limited quantity of ground truth labeling. The most important part of the medical image's detection is calculating the localization function which is normally evaluated on the Intersection over Union threshold (IoU). It helps to locate the lesion accurately to collect dominant features of the skin lesion. In this work, an object localization for skin lesion detection has been proposed using SSD- Mobilenet model on ISIC 2018 as a training and testing datasets. To evaluate the detection performance, the detection process has been achieved using two different methods; a real-time mobile application of Android camera (Galaxy S6), and Jupyter Notebook of TensorFlow Object Detection Application Program Interface (API). The total confidence scores detection quality (total mAP) is 96.04% with a total loss of 0.78. Experimental results reach 99% of detection accuracy when using Jupyter Notebook, while it reaches 100% with Android detection. The experiments have been executed on Ubuntu 16.04LTS GTX1070 @ 2.80GHZ x8 system.

Keywords: Skin lesions; Melanoma; SSD-Mobilenet; Android camera detection; TensorFlow Object Detection API

1 Introduction

Over the last decades, the rate of skin cancers exceeds the communal cancer cases for lung, breast, prostate, and colon as the skin cancer foundation had reported. In a recent medical trial, dermatologists mostly check the patients by visual analysis with physical measurements parameters which known as dermatoscope to identify the cases. Practically, it is hard to identify the type of skin lesions by bare eyes according to self-vigilance and medical check. Dermoscopy is not supposed to let imagining of skin surface by the light expanding device and immersion fluid. Usually, it is one of the highest used imaging ways in dermatology, and the diagnosis rate has been enhanced.

Nowadays, deep learning has grown so popular in medical imaging study as well as classification of skin lesions. In 2017, International Skin Imaging Collaboration led the ISBI challenge to get the best performance measures for skin lesion segmentation. Though, this challenge had focused on segmentation and skin classification, not skin lesion detection. Objects detection using a mobile camera has been applied in many purposes, like video surveillance, object stability, and collision revocation. Some techniques have been investigated to detect tracing objects from a non-stable platform. However, these techniques need to know the movement measurements of the camera, which is mostly not easy to obtain or are unable to detect an object if the size of the tracking object is small. In this work, a new detection application for ISIC 2018 skin dataset using an Android phone's camera has been presented. The software is based on Deep learning model running on TensorFlow's Object Detection API using the Android platform. A strong tool gives it straight to construct, train, and use object detection models. Generally, training a whole convolutional network from scratch consumes a lot of time and required a huge dataset. So, this problem can be fixed by using the power of transfer learning with a pre-trained model using the TensorFlow API.

Recently, an Android platform has been wildly used to the object recognition applications, which works on images taken through a built-in camera. Android is growing to be the generally used platform amongst smartphone technologies as a monitor. The user can get the correct lesion detection from the skin by which the required lesion is noticeable upon the detected skin area.

The main contributions of this paper include:

1) Proposing the use of SSD-Mobilenet for training mode on skin lesion images of ISIC Challenge 2018 dataset and monitoring the progress of our training and evaluation jobs using Tensorboard (Classification loss, localization loss, total loss, and detection accuracy).

2) Identify a candidate checkpoint to export (typically the most recent) and store it.

3) Run the detection process on a few untrained images using two different ways; by trying out the Jupyter notebook, and by developing an Android camera app, which uses the TensorFlow (TF) Detect app in mobile devices for real-time detection and localization of skin lesions.

Generally, diagnosing the dermoscopic images requires multi-stages, which include image pre-processing, image segmentation, features extraction, and classification as many of the state-of-the-art had done. The main point of those challenges was focused on segmentation and classification. A Region of interest (ROI) detection was demonstrated importance in medical image analysis. It is defined as a bounding box circumscribed the skin lesion. Moreover, by using image manipulation methods, scientists augmented the data to overcome data deficiency in the deep learning area. Tschandl et al. used different magnifications and angles that yield normal data-augmentation for HAM10000 dataset. Though, it means difficult and redundant data for the same skin lesion. Extracting 437 features was the main method in (M. Celebi 2007), which including shape, color and texture feature, leading into selecting 18 dominant features. At that point, a support vector machine (SVM) was used to recognize the lesions. While in (F. Xie 2017) three features were extracted; color, texture, and border features and a neural network ensemble model for the lesion classification. In those work, preprocessing of the original image was complex. Nevertheless, their simplification capability is weak and frequently does not have the ability to achieve a good multi-classification task.

Recently, deep learning techniques have accomplished outstanding performance in respect of classification tasks. One of the common deep learning approaches is the convolutional neural network (CNN). Basically, it is multi-layer nonlinear neural networks, which can fully exploit the inherent structure of high dimensional data and obtain dominant features from the original data naturally. In (S. Singh 2016), the data set was initially preprocessed using K-mean Clustering algorithm. This preprocessing helped to increase the rate of restructuring by eliminating all inappropriate texture. The preprocessed data was used to extract the features. In (H. Zhou 2017), a new multi-classification technique based on convolutional neural network (CNN) was used for dermoscopy images. A CNN had been trained on more information than the original remaining structure. Transfer learning had been used to train the designed network. In (M. Goyal 2018), they use two object localization meta-architectures for end-to-end ROI skin lesion detection in dermoscopic images. Faster-RCNN-InceptionV2 and SSD-InceptionV2 were trained on the ISBI-2017 training dataset and evaluate the performances on ISBI-2017 testing set, PH2 and HAM10000 datasets. It is worth to mention that our work in (Fadwa 2018) focuses on detecting human health-related actions (HHRA) from a video sequence using an Android camera. Two promising new approaches for HHRA detection has been proposed: SSD Mobilenet and Faster RCNN Resnet models. The proposed methods are estimated on the NTU RGB+D dataset.

2 Methods

In this paper, the proposed system goals to detect skin lesions by using the Android camera and TensorFlow Object Detection Notebook (Jupyter software). The dataset splits into training and testing dataset. First, all the dataset is labeled by drawing a bounding box (ground truth) around the skin lesion. Then, save them as an XML file. The XML file converts to a CSV file, and TensorFlow converts all these files into “TFRecords” to make it readable for TensorFlow application. The system uses SSD-Mobilenet as a per-trained model of TensorFlow object detection API. During training, model evaluation can be monitored to see the progress of the training and evaluation jobs using Tensorboard. Fig.1 shows the block diagram for the proposed system.

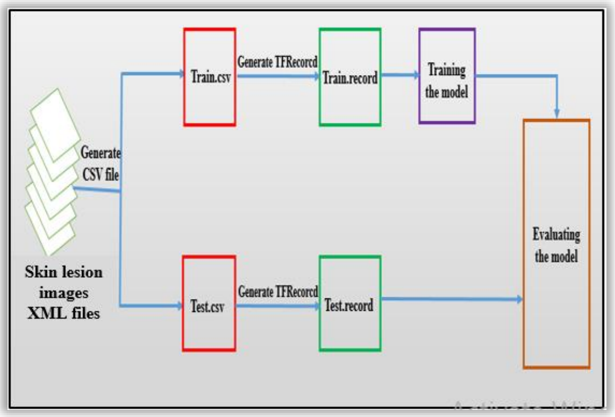

Fig .1 Basic Block Diagram for proposed the HHRA model.

2.1 Tensorflow Object Detection API

TensorFlow Object Detection API’s is a package to determine object detection applications. It is a technique of detecting real-time objects in an image. The Tensorflow Object Detection API library includes multiple out-of-the-boxes object detection constructions like SSD (Single Shot Detector), Faster R-CNN (Faster Region-based Convolutional Neural Network), and R-FCN (Region-based Fully Convolutional Networks).

In 2016, A Single Shot Multibox Detector (SSD) has been launched by investigators from Google. It is a quick single-shot object detector for several classes. Its structure is based on one feed-forward convolutional network. That leads to adopt classes straightforward and anchor stabilizer with no need to an additional stage for each proposal classification procedure. The basic feature of SSD is the employ of multi-scale convolutional bounding box outputs connected to various feature maps at the highest of the network. The main structure of SSD is extracted from VGG-16 structure. VGG-16 is applied as the core network since it has an active performance in high-quality image classification functions and transfer-learning training so as to boost the results. The bounding box method of SSD is determined by Szegedy’s project on MultiBox, and an Inception mode convolutional network is used. The loss function of the MultiBox includes two important parts that achieved their path to SSD. First is the confidence loss that estimates how confident the system in determining the bounding box. While the second part is location loss, which measures the distance of the network’s predicted bounding boxes from the ground truth ones during the training mode.

For the location loss calculation, SSD uses smooth L1-Norm. Concerning the classification process, the SSD completes object classification. Thus, for each predicted bounding box, groups of N classes predictions are determined for each possible class in the dataset. Besides, feature maps are a description of the dominant features of the image at several scales. Later, working MultiBox on multiple feature maps increases the probability of any object whether big or small that is meant to be finally localized, detected, and correctly classified.

2.2 TensorFlow in Android

The Android model of the TensorFlow library includes four applications, which they are TF Detect, TF Classify, TF Stylize, and TF Speech. All of them use the same primary code. These applications take input as a real-time image or video when the Android's camera is on. In this work, TF

Detect app has been used. The main code of the TensorFlow is written in C++ in order to build the process for Android. Besides, JNI (Java Native Interface) has been used to call the C++ functions, for instance, load-Model, obtain predictions, etc (B. Readme 2018). Two main files will be built, a shared object (.so) file, that is a C++ compiled file, and a jar file that will contain JAVA API, which calls for the native C++. Then, the JAVA API will be called to make the processes accomplished simply. Also, there are software, dependencies, and packages required: Android Studio, Android SDK and Android NDK.

2.3 Skin-Lesion Datasets and Preprocessing

Images of the skin are the most easily captured form of a medical image in healthcare, and the domain shares qualities to other standard computer vision datasets, serving as a natural bridge between standard computer vision tasks and medical applications.

For this work, we used a publicly available dataset for skin lesions that are ISIC-2018 Challenge (Skin Lesion Analysis Towards Melanoma Detection" grand challenge datasets). The lesion images come from the HAM10000 Dataset and were acquired with a variety of dermatoscope types, from all anatomic sites (excluding mucosa and nails). Every lesion image contains exactly one primary lesion; other fiducial markers, smaller secondary lesions, or other pigmented regions may be neglected. The input data are dermoscopic lesion images in JPEG format.

It is needed to have a ground truth of what exactly the object is. The ground truth is drawing a bounding box around the skin lesion as shown in Fig.2. Each lesion is tagged as a name of (skin cancer). LabelImg software saves the annotations as XML-files in PASCAL VOC format, which is prepared for creating TFRcords (Tensor Flow record format). Each dataset requires a label map connected with it, which represents a mapping from string class names to integer class IDs. Label maps should always start from ID1. In this work, there is one ID related to skin cancer class. Once all skin images are labeled, the dataset splits into two different groups; training and testing dataset.

2.4 Feature Extraction

Feature Extraction is a main necessity with the purpose of representing an object. A collection of features can represent an object by as a feature vector. This feature vector is used to distinguish objects and classify them. In image classification and object recognition systems, feature extraction is a vital section. It includes mapping the image pixels into the feature space. In this work, Mobilenet has been selected as feature extractors which is the most common in computer vision area.

Mobilenet has been used for active inference in many mobile vision functions. The depth wise separate convolution is the main part of the Mobilenet network structure. It is a progressive state of the inception unit, where parted spatial convolution is used to each channel. The 1x1 convolution with all the channels is applied to combines the output designated as pointwise convolutions. Consequently, the separation in depth wise and pointwise convolution rise the efficiency performance also. It improves accuracy, whereas a cross-channel and spatial correlations mapping is learned autonomously. MobileNet has been presenting to reach an accuracy equal to VGG-16 on ImageNet with select 1/30th of the design cost and model size. Its internal structure set is depth wise separable convolutions that factorize a typical convolution for a depth wise convolution, and a 1 × 1 convolution eventually. Practically, that would reduce the couple of the calculation cost and the number of hyper-parameters.

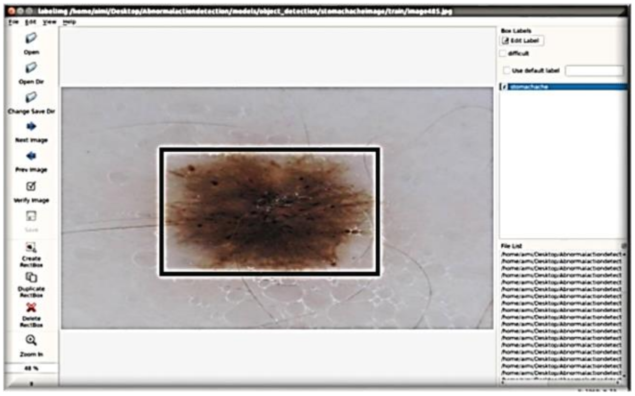

Fig. 2 Sample of skin lesion surrounded by a bounding box

The setting of TensorFlow object detection API for SSD-Mobilenet model has been adjusted the weights by an abbreviate normal distribution with a standard deviation of 0.03. The momentum _optimizer_value is 0.9 with batch size equals 32, and the intial_learning_ rate is 0.003 with a learning_ rate_ decay of 0.9997. A loss function uses the sigmoid function for classification loss, while the localization loss uses the smooth L1 function. Also, ReLU function is used as the activation function.

2.5 Testing (Detection)

To authorize the model's performance, many investigations have been accomplished on the ISIC 2018 datasets. In this work, the detection process is implemented in two ways; TensorFlow Object Detection Notebook and the Android camera. To perform the detection process in both ways, the model must be exported as a static inference graph trained on the skin lesions, as well as the corresponding label map. The TensorFlow object detection API library provides the script, which named export_inference_graph with using the latest checkpoint number at the last step that stopped the training process. It has used the 16.04LTS GTX1070@2.80GHZ x8 system to run the object detector.

The loss functions for the model include two functions; classification and localization as following:

u: True class label, $u \in 0,1, \ldots, K$; by convention, the catch all background class has u=0.

p: : Discrete probability distribution (per RoI) over k+1 classes: $p = (p_0, \ldots, p_k)$ computed by a softmax over the k+1 outputs of a fully connected layer.

v: True bounding box $v = (v_x, v_y, v_w, v_h)$.

t^u : Predicated bounding box correction, $t^u = (t_x^u, t_y^u, t_w^u, t_h^u)$.

The loss function sums up the rate of classification and bounding box prediction: $\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box}$. For "background" RoI, $\mathcal{L}_{box}$ is avoided by the indicator function $1[u \geq 1]$, described as:

$$1[u \geq 1] = \begin{cases} 1 & if\ u \geq 1 \\ 0 & otherwise \end{cases} \tag{1}$$

The overall loss function is:

$$\mathcal{L}(p, u, t^u, v) = \mathcal{L}_{cls}(p, u) + 1[u \geq 1]\, \mathcal{L}_{box}(t^u, v) \tag{2}$$

$$\mathcal{L}_{cls}\ (p,u) = -log\, p_u \tag{3}$$

$$\mathcal{L}_{box}(t^u, v) = \sum_{i\epsilon\,\{x,y,w,h\}} L_1^{smooth}\,(t_i^u - v_i) \tag{4}$$

The bounding box loss L_{box} measure the difference between t_i^uandv_i applying a robust loss function.

2.6 Quantitative Performance Measures

2.6.1 Evaluation metrics

The most important evaluation metrics in our work are precision, recall, F1-score and mean average precision (mAP). Precision shows how valid detection outcomes are, (Eq.5) indicates these metrics as following:

$$Precision = TP/((TP + FP) \tag{5}$$

where TP = true positive, FP = false positive.

Recall: represents the percentage of objects, which are detected including the detector. (Eq. 6):

$$Recall = \ TP/((TP + FN) \tag{6}$$

where FN = false negative.

2.6.2 Intersection over Union

Intersection over Union (IoU) is an accuracy estimation parameter showing how the object detector is accurate on a related database. For object localization, it is essential to fix the system prediction about the location of the exact object that is supposed to be detected as shown in Fig.3. Usually, this also includes sketching a bounding box around the object of interest. Then, the localization function is estimated on the Intersection over Union threshold (IoU).

The TensorFlow Object Detection API uses “PASCAL VOC 2007 metrics”, and a location estimated instance is defined as a TP, whereas Intersection over Union (IoU) is above 50%. One object can be related to one bounding box at a time. Though, if some bounding boxes are predicted for an object, one is viewed as TP and the others as FP. Nevertheless, if an object is with no predicted bounding box, which is related to it, it is predictable as an FN.

2.6.3 Mean Average Precision (mAP)

The mAP is the result of precision and recalls “precision-recall” calculations on determining bounding boxes. It is a suitable means of how good the network to choose the objects of interest. The larger the score of the mAP, the more precise the detection system. Mostly, the mAP improves “precision-recall” curve. Then, for each guess, a recall rate and a precision rate is determined. The average precision (AP) is the predictions’ average estimated over several thresholds. It is a determination to reach characteristics of the detector in a single number. In this work, the AP curves for the skin lesions class based on the pertrained models (SSD-Mobilenet) have been shown in Figs.4-5. A vertical axis denotes the AP values, whereas a horizontal axis denotes the steps (epochs).

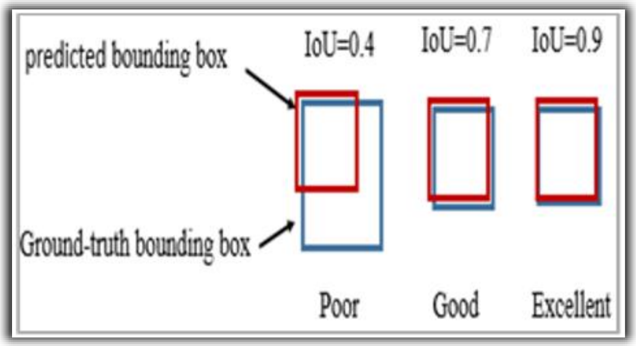

Fig. 3 An instance of calculating IoU for different bounding boxes.

3 Results

3.1 Training and Evaluation

TensorFlow provides many configuration documents which can be reformed according to a new training set. A TensorFlow’s imagining platform, named as TensorBoard, can be used to observe the results of the training and the evaluation stages. It can observe several parameters, for instance, the training time, total loss, number of steps and much more. In addition, TensorBoard can be run during the system training mode, which allows you to monitor the progress of your training and evaluation jobs to check that the training performance is going in the correct way. The system checkpoint file is applied as a start point for the fine-tuning process.

Throughout the evaluation, the mAP indicates the trained model ratio of accurate predictions for our label. While the IoU is specific to object detection models. This measurement signifies the overlap between the bounding box and the ground truth-bounding box, defined as a percentage. The mAP graph is an average of the percentage of correct bounding boxes and labels of the model restarted with “correct” relating to bounding boxes that had 50% or more overlap with their corresponding ground truth boxes. During the training developments, the main aim is to decrease total loss as minimum as can be (less than 1). Tensorboard diagrams for AP, mAP, classification loss, localization loss, and total loss for SSD-Mobilenet model are shown in Figs 4, 5, 6, 7 and 8. The training process should be completed when the total loss reduces as the iterations/steps(epochs) are increasing. The parameter num_steps describe how many training steps needed before ending the training process. It based on the dataset size with how long the investigator desires to do the training of the model. The applied metric is a mean average precision (mAP). It is a single number represents the area under the precision-recall curve. The mAP is a measure of how good the model in generating a bounding box that has at least a 50% intersection with the ground truth bounding box in the test dataset. The mAP value got to higher confidence at 0.5IoU. The higher the mAP values, the higher the detector performance.

The classification loss curve in Fig.6. show the validation of skin lesion class which is classified and matched with the earlier trained class. The lower the classification loss that decreases to zero, the higher the classification and detector accuracy. Fig.7. shows the result of the localization loss curve. It defines the predicate bounding box that matches with the ground-truth bounding box. Table 1 reviews all the parameters (mAP, Total loss, Classification loss, Localization loss) of the training and evaluation processes, and the number of the steps with the consumed time to achieve these requirements.

Table 1. Benchmark results of the cascade oscillators model

Steps	mAP	Classification loss	Localization loss	Total loss
8.622k	0.96	-	-	-
11.62k	-	0.45	0.48	0.78

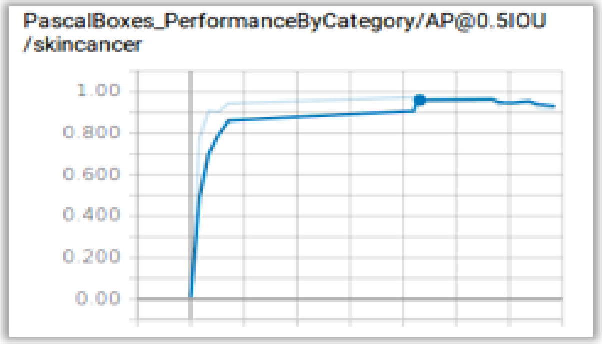

Fig. 4. Maximum AP curves at 0.5 IoU for skin cancer lesion at 8.622k steps.

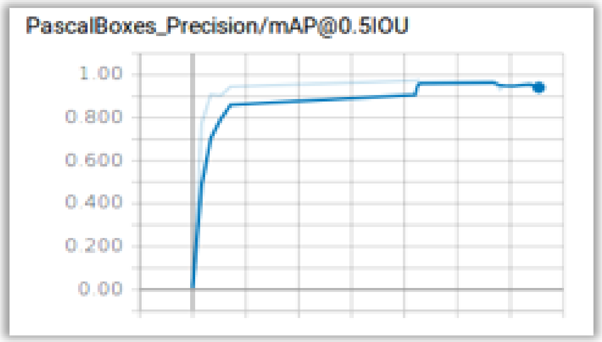

Fig. 5. Maximum mAP at 0.5 IoU at 8.622k steps.

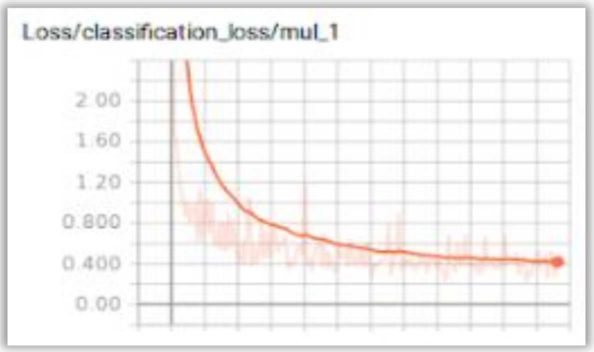

Fig. 6. Minim classification loss for skin cancer class.

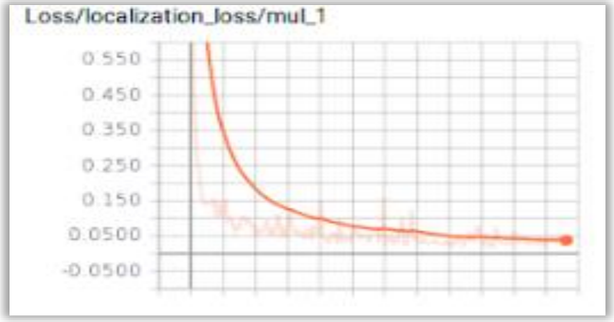

Fig. 7. Minim localization loss for skin cancer class.

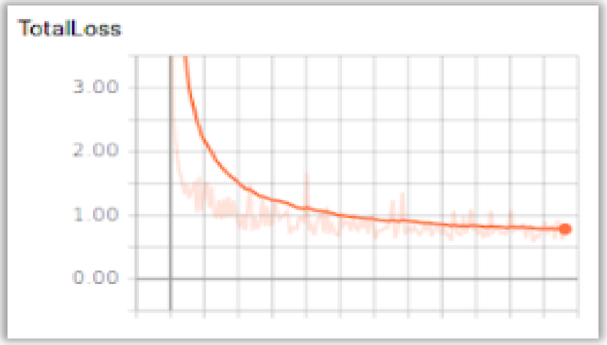

Fig. 8. Minim total loss for skin cancer class.

3.2 Detection Results

To authenticate the detection performance for skin cancer detection, inclusive investigations have been completed on the datasets. To achieve the highest expectable detection accuracy, groups of skin lesion images are tested. In this work, the detection process is executed using two different ways; TensorFlow Object Detection Notebook and the Android camera. To do the detection process in both ways, the system must be exported as a static inference diagram trained on the skin lesion dataset, along with the equivalent label map. The TensorFlow object detection API library delivers the script, named export_inference_graph using the newest checkpoint number at the last step which the training process has been stopped. It has been used the 16.04LTS GTX1070@2.80GHZ x8 system to run the object detector on ISIC 2018 dataset.

3.2.1 TensorFlow's Object Detection Notebook

When the supplies are finished, the detection process is done by using TensorFlow's Object Detection Notebook. The detection results are exposed in Fig.9. The results have shown such a high detection accuracy (99% and 100%).

3.2.2 TensorFlow in Andriod

The model will be exported to an Android phone (Galaxy S6). The testing has been done by capturing the skin lesion from the phone's camera and bounding detection boxes with the name of the skin cancer class and detection percentage accuracy as described in Fig.10.

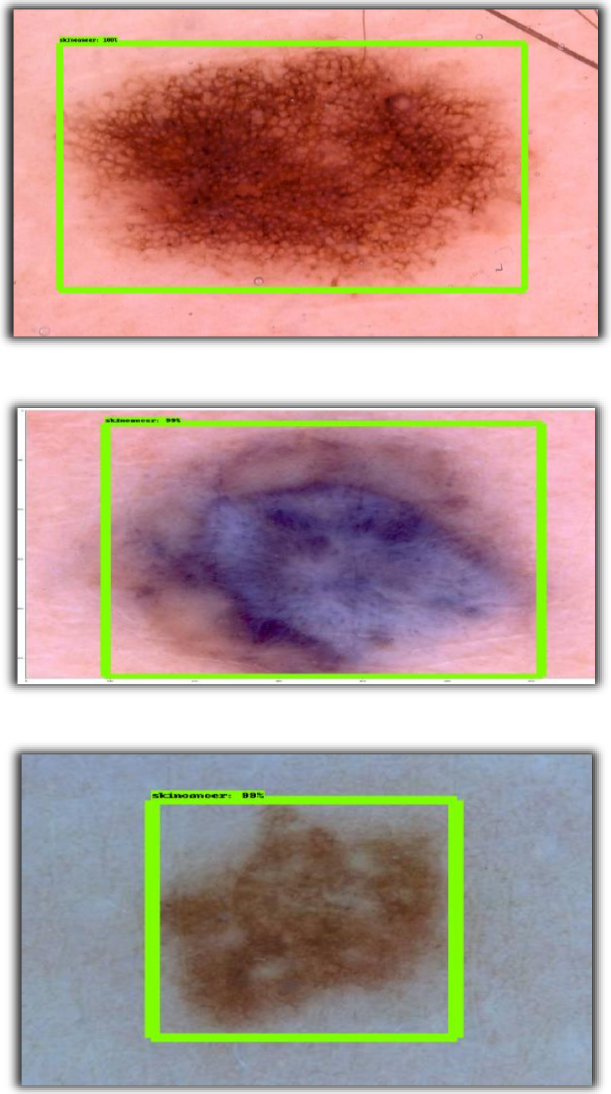

Fig. 9. Samples of the detection results that used TensorFlow object detection notebook technique.

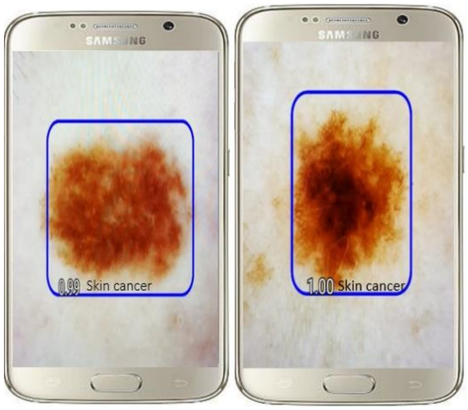

Fig. 10. Samples of the detection results using Android camera.

4 Discussion

In this paper, skin cancer lesions have been detected using TensorFlow object detection API system. For training the dataset, a new pre-trained model (SSD-Mobilenet) has been used. The average precision (AP) is the average of class predictions calculated over numerous thresholds. While the detection accuracy (mAP) is a high rate with different num_steps at 0.5IoU. This is because that the system deals with the skin images, so it needs a long time to train the whole samples of each image. The num_steps number directly affects the size of the training dataset. Localization loss denotes to the predicted bounding box when matches the ground-truth bounding box. Nevertheless, when the loss cost is low, and the Intersection over Union is high, that indicates that the bounding box is in the exact location of the lesion. The classification loss illustrates the efficiency of skin lesion class which is matched with the prior trained class. The lower the classification loss values, the perfect the classification accuracy is, which leads to the high-level performance of the detector. The Android detection results and the detection results of TensorFlow Object Detection Notebook have shown high performance.

5 Conclusion

In Conclusion, a new detector system was settled for skin cancer lesions using two methods; TensorFlow object detection API notebook, and TensoFlow in Android using the phone's camera. The used data was extracted from the International Skin Imaging Collaboration “ISIC 2018”. The detection model was trained and estimated based on SSD-Mobilenet in the training mode. According to the results, the mAP total achieved 96.04%, and the lowest error was calculated through both losses of classification and localization and it was less than 0.5. Besides, high-performance efficiency has been enhanced by using the smartphone. As future research, we plan to use Cloud computing to reduce training and evaluation time.

References

Goyal, M., Member, S., and Yap, M. H. (2017) Region of Interest Detection in Dermoscopic Images for Natural Data-augmentation, XX(X), 1–6.

Esteva, A. (n.d.) Dermatologist-level classification of skin cancer with deep neural networks.

Codella, N. C. F., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., … Heights, Y. (2017) No Title, 1–5.

Sayali, J., and Pereira, D. (2015) Object Recognition In Mobile Phone Application For Visually Impaired Users, 2(2), 17–20.

Celebi, M. E., Iyatomi, H., and Stoecker, W. V. (2009) Approximate Lesion Localization in Dermoscopy Images, (March 2019). https://doi.org/10.1111/j.1600-0846.2009.00357.x

Goyal, M., Member, S., and Yap, M. H. (2018) Region of Interest Detection in Dermoscopic Images for Natural Data-augmentation, (July). https://doi.org/10.13140/RG.2.2.16602.64966

Tschandl, P., Rosendahl, C., and Kittler, H. (2018) Data Descriptor: The HAM 10000 dataset , a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Nature Publishing Group, 5, 1–9. https://doi.org/10.1038/sdata.2018.161

Imaging, C. M., Celebi, M. E., Kingravi, H. A., Iyatomi, H., and Stoecker, W. V. (2007) A Methodological approach to the classification of dermoscopy images A methodological approach to the classification of dermoscopy images, (October). https://doi.org/10.1016/j.compmedimag.2007.01.003

Xie, F., Fan, H., Li, Y., Jiang, Z., Meng, R., and Bovik, A. (2017) Melanoma Classification on Dermoscopy Images Using a Neural Network Ensemble Model, 36(3), 849–858.

Singh, S. K., and Jalal, A. S. (2016) A Robust Approach for Automatic Skin Cancer Disease Classification, (Xlm).

Al-azzo, F., Mohammed, A., and Milanova, M. (2018) Human Related-Health Actions Detection using Android Camera based on TensorFlow Object Detection API, 9(10), 9–23.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., and Berg, A. C. (n.d.). SSD: Single Shot MultiBox Detector, 1–15.

Szegedy, C., and Reed, S. (2015) Scalable, High-Quality Object Detection, (March).

Smooth, T., Rcnn, F., and Rcnn, F. (n.d.) Smooth L1 Loss, (2), 2–5.

Anar, A. C., Bostanci, E., and Guzel, M. S. (n.d.) Live Target Detection with Deep Learning Neural Network and Unmanned Aerial Vehicle on Android Mobile Device.

Codella, N. C. F., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., … Heights, Y. (2017) Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC), 1–5.

Tschandl, P., Rosendahl, C., and Kittler, H. (2018) Data Descriptor: The HAM 10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Nature Publishing Group, 5, 1–9. https://doi.org/10.1038/sdata.2018.161Howard, A. G., and Wang, W. (2012) Applications.

Everingham, M., Gool, L. Van, Williams, C. K. I., Winn, J., and Zisserman, A. (n.d.) The P ASCAL Visual Object Classes (VOC) Challenge.

Mustamo, P. (2018) Object detection in sports: TensorFlow Object Detection API case study, (January).

Karpathy, A., and Fei-fei, L. (2017) Deep Visual-Semantic Alignments for Generating Image Descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 664–676. https://doi.org/10.1109/TPAMI.2016.2598339