I.J. Information Engineering and Electronic Business, **2020, 2, 21-27**
Published Online April 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2020.02.04

Early Skin Cancer Detection Using Deep Convolutional Neural Networks on Mobile Smartphone

Justice O. Emuoyibofarhe, Daniel Ajisafe
Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso
Email: eojustice@gmail.com, ajisafedaniel@gmail.com

Ronke S. Babatunde
Department of Computer Science, Kwara State University, Malete, Kwara State
Email: Ronkebabs711@gmail.com

Meinel Christoph
Hasso Plattner Institut, University of Potsdam, Germany
Email: Christoph.meinel@hpi.de

Received: 04 November 2019; Accepted: 23 December 2019; Published: 08 April 2020

***Abstract*—**Malignant melanoma is the most dangerous kind of skin cancer. It is mostly misidentified as benign lesion. The chance of surviving melanoma disease is high if detected early. In recent years, deep convolutional neural networks have attracted great attention owing to its outstanding performance in recognizing and classifying images. This research work performs a comparative analysis of three different convolutional neural networks (CNN) trained on skin cancerous and non-cancerous images, namely: a custom 3-layer CNN, VGG-16 CNN, and Google Inception V3.

Google Inception V3 achieved the best result, with training and test accuracy of 90% and 81% respectively and a sensitivity of 84%. This work contribution is mainly in the development of an android application that uses Google Inception V3 model for early detection of skin cancer.

***Index Terms*—**Skin cancer, Convolutional Neural Networks, Medical Images, Android device.

I. Introduction

Skin cancer is one of the most common cancer in the world. More than 3.5 million new cases occur annually in the US, and these numbers continue to rise [1]. Of the three most common types of skin cancer, melanoma is the deadliest, accounting for more than 75% of all skin-cancer deaths [2]. The lethality of melanoma is directly dependent upon the stage of cancer at the time of diagnosis [3]. If diagnosed early, almost all skin cancer can be successfully treated. However, those with delayed diagnosis and advanced disease continue to have a grave prognosis [4].

There will be a demand for dermatologists, health workers, and primary care physicians (PCPs) in the near future, as a massive shortage of these professionals of more than 91,000 is expected by 2020 [5] and 12.9 million by 2035 [6]. On the contrary, the number of mobile phone users in the world is expected to increase to 4.78 billion [7] by 2020.

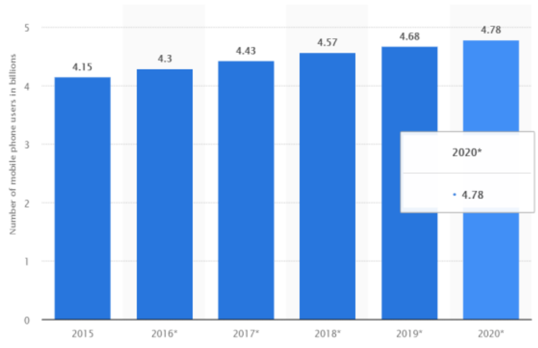

Fig. 1. No of mobile phone users from 2015 to 2020

There is need to develop intelligent mobile solutions to assist dermatologists and health professionals in early detection of this deadly disease and make better decisions about a patient's health especially in low-income countries. Therefore, the objective of this paper is to develop and evaluate the use of CNN for early detection of skin cancer.

II. Related Works

Some dermatologists have implemented the ABCDEs that explains the attributes of malignant lesions [8]. The ABCDEs is an abbreviation that stands for Asymmetric shape, Border irregularities, Color, Diameter and Evolution over a period of time. However, these attributes can be misleading and not all melanoma fall within these parameters. Due to different factors, the results are sometimes unreliable.

There has been a number of machine-learning based approaches employed in classifying skin diseases. These approaches include support vector machines to artificial neural networks. Recently, some studies [9, 10] have used the fine-tuned VGG 16 architecture but failed to integrate an end-user application.

Lubax (Lubax Inc, CA, USA), [11, 12, 13] is one few systems working on mobile platforms for cancer detection. However, many of these systems only used the mobile device for capturing, storing and transmission of the skin lesion images to a remote server without performing any kind of computation locally on the mobile phone. For example, [12] and [13] capture images using the derma scope attached to the mobile device and send the images to the server for computer evaluation.

This research work performs a comparative evaluation of three different CNN architectures and proposes a light-weight skin lesion-detection android application running locally on the mobile phone.

III. Methodology

A. Datasets

The training data was acquired from the International Skin Imaging Collaboration (ISIC) archive [14]. Images in the archive were publicly made available by the melanoma project group to facilitate the application of digital skin imaging to reduce melanoma-related death rates. A balanced dataset of 2358 dermoscopic images was used that includes 1179 melanoma images and 1179 benign images. The dataset was split into training and test data in a 70 to 30% ratio. The training data included 1758 labeled images and the test data included 600 labeled images.

Table 1. Class/ labels

	Benign	Malignant	Total Images
Training set	879	879	1758
Test set	300	300	600

The images were arranged in a well-defined directory containing the training and test folders. Each folder had sub-folders that represented both classes, benign and melanoma.

B. Data augmentation

It is expected that exposing the model to different variations of images would increase its learning outcome and make it generalize well. Therefore, we carried out a series of image transformations on the training data during training. These included rotations, horizontal flipping, vertical flipping, image zooming, and shearing. The transformation feature comes with keras library using the image data generator method. As a golden rule, the test data was not augmented in order to get the true performance of the models.

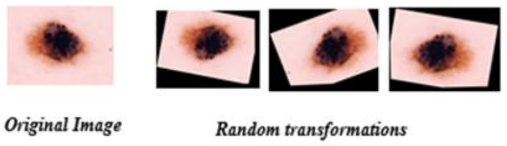

Fig. 2. Image transformations

C. Image Pre-processing

Feature scaling is an important technique for machine learning algorithms. Gradient descent converges faster when it is applied. The range of pixel values for features of our input images varied largely, each pixel can fall anywhere in between 0 - 255. The dataset also contained few images with poor contrast. Min-max normalization was applied by dividing each pixel value by 255. After normalization, the pixel values were within the scale of 0 and 1. Re-scaling also prevented potential issues caused by images with poor contrast. Finally, the images were resized to the expected size of 224x224 for each of the three architectures.

$$x = x / 255 \; . \qquad (1)$$

Where X is a single pixel value

D. Frameworks used

Python is the core programming language used to develop this project and responsible for most of our scripting code. Keras is a simple, high-level neural networks library, written in python that works as a wrapper to either Tensorflow or Theano. It is a high-level API used to build and train deep learning models. It is known for its advantage for fast prototyping, advanced research, and production over its contemporaries. Hence, the reason why we chose the framework for this project. The CNN architectures were implemented using python and Keras with Google's Tensorflow as the backend.

E. Environment setup

All model training for this project was carried out on kaggle kernels. Kaggle offers free access to Nvidia K80 GPUs in its kernels. This has 5x speed than running the same models on a CPU. Our keras deep learning framework leveraged this improved speed. However, its GPU instances can only run up to 6hours before timing out. Code development was done using Anaconda jupyter notebooks locally and trained on the kernel notebooks. The models are saved in keras .h5 file format which stores both the architecture and weights using the model_save function.

F. Environment setup

Transfer learning is a method in deep learning where a model developed for a task is reused as the starting point for a model on another task. Due to the relatively small number of images of a skin lesion in our dataset, this method initializes a new model with weights from a model pre-trained on a larger dataset such as ImageNet [15]. The process is used for architectures that are computationally expensive to train like VGG16 and Inceptionv3. The underlying assumption behind transfer learning is that the pre-trained models have already learned features that are useful for the image classification task at hand.

IV. CNN ARCHITECTURES AND RESULTS

This section details each network architecture, choice of optimizers, loss function, hyper-parameters, no of epochs, learning rate and model schema.

A. 3-layer Convolutional Neural Network

This is a custom CNN with 3 convolutional layers and 2 fully connected layers. Each convolutional layer is followed by a max pooling layer with a pool size of 2 to provide invariance to shifts in position and improve generalization performance. We tried RMSprop as our optimizing function but discovered that the model was not learning. Accuracy juggled around 50%, no better than guessing randomly. The learning rule was changed to Adam optimizer and there was an improvement in performance. We applied dropout as a regularization technique to the fully connected layers to prevent over-fitting and found the rectified linear unit (RELU) activation function to work best for this model. Since we have two classes, our loss function is a binary cross entropy calculated as:

$$-(y\log(p) + (1-y)\log(1-p)) \quad (2)$$

The architecture was initialized with random weights and trained for 20 epochs. After each epoch, the model learns new features and computes new weights using backpropagation. The classifier achieved an accuracy of 68% without performing data augmentation during training. With augmentation, accuracy improved to 72% in less than 10 epochs. There was no significant increase after the 10th epoch. However, it served as a baseline for comparison against the other two architectures.

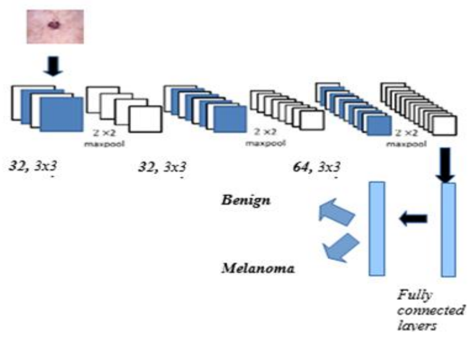

Fig. 3. 3-layer Convolutional Neural Network

B. VGG 16

The VGG16 network was invented by the visual geometry group from where its name was coined and was the winner at ILSVRC (ImageNet Large-Scale Visual Recognition Competition) 2014 localization task and 2nd best in the classification task. It is a very deep, 16-convolutional-layer network with a uniform architecture. The model was originally trained on millions of images in a data store called ImageNet [15]. It uses only 3x3 same convolutions with an increasing number of filters. The architecture consists of 5 convolutional blocks (a total of 13 convolutional layers) and a fully-connected classifier (a total of 3 fully-connected layers). Pooling layers are not counted as formal layers as they do not have trainable parameters or weight matrix. We instantiated the pre-trained VGGNet from the 1st convolutional block up to the last and loaded its weights. We then pre-computed the bottleneck features since that is where most of the computation time is. Once we pre-computed them, the activation maps were used to train a small fully-connected model on top of the bottleneck features. RELU was used as our activation function for the hidden layers and sigmoid at the output layer.

$$\text{Max}(0, Z) \qquad \text{RELU} \quad (3)$$

$$\sigma(X) = 1/(1 + \varepsilon(-X)) \qquad \text{SIGMOID} \quad (4)$$

The trained top model was then added on top of the convolutional model in order to fine-tune the last convolutional block and our trained classifier. We compiled the final model using stochastic gradient descent (SGD) optimizer with a very slow learning rate of 11e-4 and momentum of 0.9 in order not to ruin the pre-trained weights. The model achieved an accuracy of 74% without data augmentation and 77% accuracy on augmented data after 50 epochs of training.

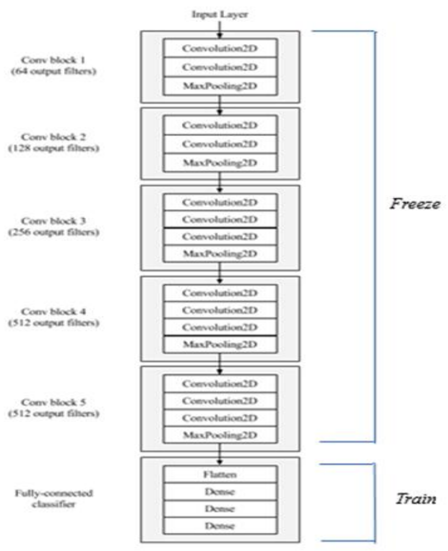

Fig. 4. VGG16 architecture: source [16]

C. *Inception V3*

The first version of the inception network, Inception v1 was created to solve the problem of variation in size and location of salient parts that are important in an image. With different filter sizes on the same level, it uses the trick of going “wider” rather than “deeper. GoogLeNet as it is called won the ILSVRC [15] 2014 classification challenge. The 3rd version, Inception v3 includes all the benefits of v1 and updates made in v2 in addition to label smoothing to prevent over-fitting, batch normalization in its auxiliary classifiers and factorized 7x7 convolutions to lessen the computational burden. We used the Inception V3 pre-trained model originally trained on the ImageNet dataset. The process of fine-tuning the network is mostly similar to VGG16 but with subtle differences. Unlike VGG16, it uses a functional API that offers more flexibility in building complex models with multiple inputs and outputs. With the model instantiated, we added a global spatial average pooling layer, a fully-connected layer (with RELU activation) and a logistic layer (with sigmoid activation) as the top model. This top model was compiled with RMSprop optimizer and trained for a few epochs. With the trained model, we could fine-tune convolutional layers from Inception v3. The model was recompiled for the changes to be affected, using SGD with a low learning rate of 1e-4 and momentum of 0.9. Finally, we trained the model once more, this time fine-tuning the top 2 inception blocks with the top fully connected layers. The model achieved 76.5% accuracy on un-augmented data and showed significant improvement on augmented data. The best test accuracy increased to 81% after 50 epochs.

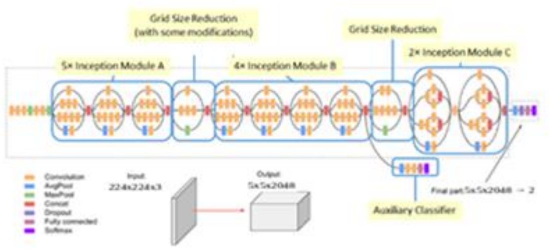

Fig. 5. Inception v3 architecture: source [17]

We carefully monitored the models during training to understand the behavior of each of them. Furthermore, they were evaluated using different metrics such as accuracy, loss, sensitivity, precision, receiver operating characteristics area under the curve (ROC AUC) score, and confusion matrix.

Training and testing results for each model without data augmentation are shown in Tables 2 and 3.

Table 2. Result of the training set

	Loss	Accuracy	Sensitivity	Precision
3_CNN	**0.10**	**96.50%**	**94.86**	**98.08**
VGG16	0.14	96.14%	94.42	97.78
IV3	0.35	84.71%	81.57	87.04

Table 3. Result of the test set

	Loss	Accuracy	Sensitivity	Precision
3_CNN	2.38	67.50%	65.67	68.17
VGG16	0.78	73.82%	**76.01**	72.84
IV3	**0.58**	**76.48%**	72.00	**78.97**

Training and testing results for each model with data augmentation are shown in Tables 4 and 5.

Table 4. Result of the training set

	Loss	Accuracy	Sensitivity	Precision
3_CNN	0.37	83.57	73.71	**91.81**
VGG16	0.25	86.50%	86.03	86.92
IV3	**0.23**	**90.00%**	**89.51**	90.30

Table 5. Result of the test set

	Loss	Accuracy	Sensitivity	Precision
3_CNN	0.94	71.17	67.67	72.76
VGG16	0.75	76.67%	78.00	75.97
IV3	**0.49**	**81.00%**	**84.33**	**79.06**

The VGG16 model achieved 96% accuracy during training and correctly identified unseen melanoma images (sensitivity) in the test data 76% of the time. However, with more data, Inception v3 produced the lowest generalization error (loss) of 0.49, highest sensitivity of 84.33 and best test accuracy of 81%. Also, the results show that more data influenced learning outcomes positively and data augmentation improved generalization performance for all models.

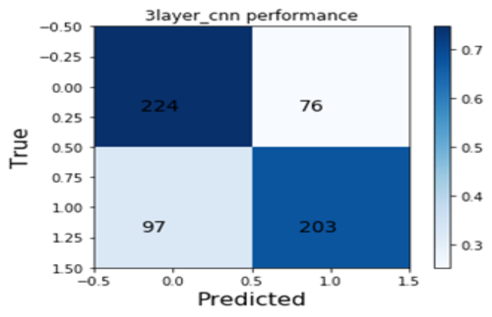

Fig. 6. 3_CNN Confusion Matrices

The Figures 6 - 8 are the confusion matrices generated by the three models on the test data with False Negatives descending with each model. Each confusion box contains the true negatives (left top), false positives (right top), false negatives (left bottom) and true positives (right bottom).

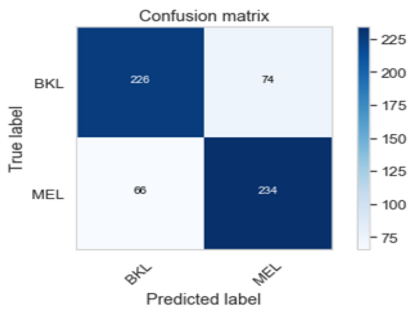

Fig. 7. VGG16 Confusion Matrices

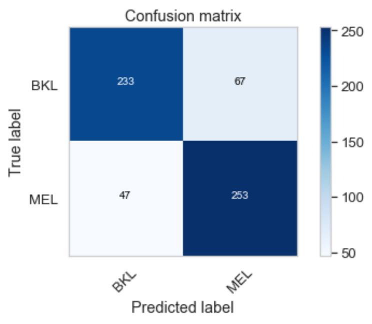

Fig. 8. Inceptionv3 Confusion Matrices

Figures 9 – 11 are the receiver operating characteristics area under curve (ROC AUC) generated by the three models. A perfect classifier has AUC = 1 and a random classifier has AUC = 0.5.

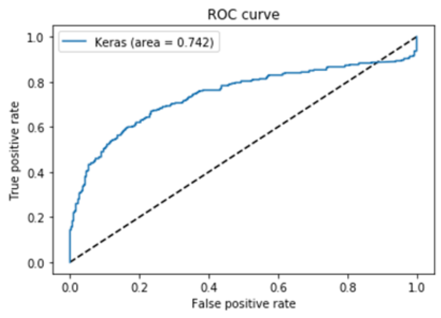

Fig. 9. 3CNN ROC

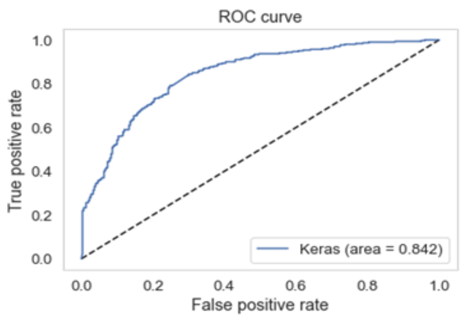

Fig. 10. VGG16 ROC

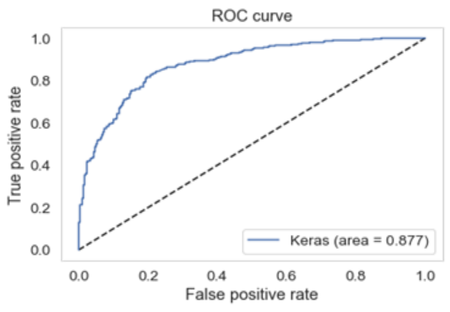

Fig. 11. Inceptionv3 ROC

Inception v3 had the lowest minimal loss, highest sensitivity and best test accuracy. In the medical domain, sensitivity is considered an important metric. Sensitivity is the percentage of true positives (melanoma cases) that are correctly identified. Based on this domain knowledge, Inception v3 outperforms its counterparts with a sensitivity of 84%.

V. Model Deployment to Mobile Application

A. *Converting Keras model to Tensorflow model*

The Google Inception V3 model in keras format with its structure is converted to a ready-for-inference tensorflow (TF) model with the keras_to_tensorflow converter [18]. This is a possibility since Keras is only a wrapper to TF. This tool converts all TF variables to TF constants and saves the inference graph and weights into a binary protobuf (.pb) file. The resultant sensor flow model holds both the model architecture and its associated weights.

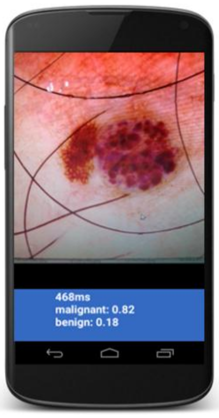

Fig. 12. Mobile application classifying a dangerous skin lesion as malignant

B. *Adding TFmodel to android*

Deploying TFmodel to mobile can be done in two ways, using TensorFlow Mobile or TensorFlow Lite. TFlite is the official solution for running machine learning models on mobile and embedded devices. It enables on-device machine learning inference with low latency and a small binary size on Android, iOS, and other operating systems [18]. However, it is still in developer preview and does not support all operations of tensorflow yet. On the other hand, Tensorflow mobile is more stable and has more support functionality. Using the latest version of Android studio, TFmobile was added as a dependency in the gradle file before being synchronized. The converted model.pb and our defined class labels were also added to the assets directory. Prediction is run on a background thread to avoid running out of memory. The application loads an image to be predicted from the file system performs prediction and displays the predicted classes with their confidence scores.

Illuminations, occlusions and image distortions remain core challenges in computer vision. Pictures with heavy distortions and large variations in lightning can mislead the classifier. It is beneficial that images to be fed into the mobile application are well-taken.

VI. Conclusion

In this paper, we built a two-class classifier that predicts whether an unseen image of a skin lesion is melanoma or benign using different methods. Of the three models implemented, Google Inception V3 achieved 81% accuracy and 84.33% sensitivity. With these high scores, deploying CNN with Google Inception V3 will complement the efforts of dermatologists and PCPs when diagnosing skin lesions. Future work could include further fine-tuning of the Google Inception V3 network to obtain improved performance metrics and also augmenting with diverse skin cancer dataset.

References

[1] Rogers, H. W., Weinstock, M. A., Harris, A. R., Hinckley, M. R., Feldman, S. R., Fleischer, A. B., & Coldiron, B. M. (2010). Incidence estimate of nonmelanoma skin cancer in the United States, 2006. Archives of dermatology, 146(3), 283-287.

[2] American Cancer Society. Cancer Facts & Figures 2013. Available at: http://www.cancer.org/acs/groups/content/@epidemiology surveilance/documents/document/acspc-036845.pdf. Accessibility verified April 8, 2014.

[3] Balch, C. M., Gershenwald, J. E., Soong, S. J., Thompson, J. F., Atkins, M. B., Byrd, D. R. Buzaid, A.C., Cochran, A.J., Coit, D.G., Ding, S. and Eggermont, A.M. (2009). Final version of 2009 AJCC melanoma staging and classification. Journal of clinical oncology, 27(36), 6199.

[4] Jemal, A., Siegel, R., Xu, J., & Ward, E. (2010). Cancer statistics, 2010. CA: a cancer journal for clinicians, 60(5), 277-300.

[5] AAMC Center for Workforce Studies, June 2010 Analysis. Available at: http://www.aamc.org/data. Accessibility verified April 4, 2014.

[6] World Health Organization (WHO). Third Global Forum on Human Resources https://www.who.int/mediacentre/news/releases/2013/health-workforce-shortage/en/

[7] Statista, https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

[8] Tsao, H., Olazagasti, J. M., Cordoro, K. M., Brewer, J. D., Taylor, S. C., Bordeaux, J. S. Chren, M.M., Sober, A.J., Tegeler, C., Bhushan, R. and Begolka, W.S (2015). Early detection of melanoma: reviewing the ABCDEs. Journal of the American Academy of Dermatology, 72(4), 717-723.

[9] Esteva, A., Kuprel, B., & Thrun, S. (2015). Deep networks for early stage skin disease and skin cancer classification. Project Report, Stanford University.

[10] Liao, H. (2016). A deep learning approach to universal skin disease classification. University of Rochester Department of Computer Science, CSC.

[11] Wadhawan, T., Situ, N., Rui, H., Lancaster, K., Yuan, X., & Zouridakis, G. (2011, August). Implementation of the 7-point checklist for melanoma detection on smart handheld devices. In Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE (pp. 3180-3183). IEEE.

[12] Gu, Y., & Tang, J. (2014, May). A mobile system for skin cancer diagnosis and monitoring. In Mobile Multimedia/Image Processing, Security, and Applications 2014 (Vol. 9120, p. 912009). International Society for Optics and Photonics.

[13] Abuzaghleh, O., Faezipour, M., & Barkana, B. D. (2015). Skincure: An innovative smart phone-based application to assist in melanoma early detection and prevention. arXiv preprint arXiv:1501.01075.

[14] "International Skin Imaging Collaboration: Melanoma Project Website," https://isic-archive.com/.

[15] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252.

[16] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[17] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[18] Amir Abdi, "keras to tensorflow converter", Github https://github.com/amir-abdi/keras_to_tensorflow

[19] Tensorflow Lite for mobile and embedded devices https://www.tensorflow.org/lite/

Authors' Profiles

Justice Emuoyibofarhe is a Professor of Computing at Ladoke Akintola University of Technology. He received his PhD in 2004. He specialises in neuro-fuzzy computing computational optimisation. He had post-doctoral fellowship at the Centre of Excellence for Mobile e-service, University of Zululand, South Africa in 2006. He is a member of the IEEE Computational Intelligence Society. He is also a Visiting Researcher at the Hasso Plattner Institute, University of Potsdam, Germany. His present research area is in the application of mobile computing and wireless communication to e-health and telemedicine.

Daniel Ajisafe is currently a Masters student at the African Masters in Machine Intelligence Program funded by Google and Facebook. He worked as a Data scientist at KPMG Nigeria where he used machine learning and advanced analytical algorithms to draw insights from structured and semi-structured data. His research interest is in computer vision and machine learning applications in healthcare.

Ronke Seyi Babatunde is an academic staff in the Department of Computer Science, College of Information and Communication Technology Kwara State University Rd, Malete, Nigeria

Christoph Meinel is a German Scientist and a University Professor of Computer Sciences. He is President and CEO of the Hasso Plattner Institute (HPI) for IT Systems Engineering at the University of Potsdam (Germany), and a Professor for Internet Technologies and Systems. Besides his teaching activities in Potsdam, he is an Honorary Professor at the Technical University of Beijing (China), a Visiting Professor at the Shanghai University (China), and a Senior Research Fellow of SnT at the University of Luxembourg. He is a Chairman or a member of various international scientific boards and program committees, and has organised several internal symposia and conferences.

How to cite this paper: Justice O. Emuoyibofarhe, Daniel Ajisafe, Ronke S. Babatunde, Meinel Christoph, " Early Skin Cancer Detection Using Deep Convolutional Neural Networks on Mobile Smartphone", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.12, No.2, pp. 21-27, 2020. DOI: 10.5815/ijieeb.2020.02.04