International Journal of Research Publication and Reviews, Vol 4, no 4, pp 5156-5161 April 2023

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SkinGuard App for Skin Cancer Detection using Neural Networks

Prathmesh Bhandarkar [a], Mansi Dharpure [b], Leena Patil [c], Kundan Chaudhari [d], Himanshi Bobde[e], Jitesh Patle [f]

[a] *Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur 440019, India*
[b] *Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur 440019, India*
[c] *Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur 440019, India*
[d] *Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur 440019, India*
[e] *Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur 440019, India*
[f] *Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur 440019, India*

A B S T R A C T

Considering that the skin is the largest organ in the human body, skin cancer is among the worst types of cancer. Genetic disorders that promote the formation of abnormal skin cell types in humans are a serious concern. Skin lesions are difficult to visually scrutinize and manually examine, which makes detecting skin cancer difficult. It is well known that the broad adoption of medical apps is driven by mobile health, a disruptive force in the industry. Machine learning has become a potent technique for categorizing medical images to identify different illnesses, increasing the system's storage and processing needs. To address this problem, several studies have focused on using cloud-based machine learning, which uses an internet connection to outsource computation that needs much more data. Unfortunately, this method has privacy and latency difficulties, especially when dealing with sensitive data. We propose an on-device inference App and use a dataset of skin cancer pictures to demonstrate proof of concept for resolving these concerns using apps. On a mobile device, we pre-trained a sort of CNN model using 10,015 images of skin cancer. The test data is kept on the device because all computations are done locally when new test images are shown, and the inference takes place locally. With this approach, latency suffers, less bandwidth is used, and privacy is enhanced [4] [5] [6].

Keywords: Machine Learning, SkinGuard, CNN, Image Classification, Convolutional Neural Networks, Deep Learning, Skin Cancer, Health Informatics

1. Introduction

Machine learning has demonstrated outstanding effectiveness in recent years in the classification of medical pictures to identify a wide range of illnesses [3] [4]. This powerful technology has been applied to mobile health, an area of medicine where mobile devices and machine learning are combining to become more and more popular. SkinGuard is widely recognized as a transformative factor in the supply of ubiquitous health informatics through medical apps. Our way of life is changing at an unheard-of rate because of SkinGuard programs that use machine learning [1] [2]. The ability to detect cancer and its many forms using machines has the potential to pave new research avenues and drastically reduce the need for laborious duties. As skin cancer prevalence climbs, it has become more crucial than ever to recognize cases early in order to preserve victims. This study includes an examination of various strategies for distinguishing between numerous kinds of malignancies of the skin. The entire machine-based processing process is completed using techniques from image processing, computer vision, and machine learning. In order to train a classification model using supervised machine learning, an abundance of information is required to be gathered. This created significant storage and processing issues for mobile apps. As a result, many mobile health apps make use of cloud-based machine learning models. Ruiz-Zafra et al., for example, provided a platform for cloud-based mobile health applications used for patient monitoring, remote diagnostics, and data collection. Gatsios et al. used mobile technologies, cloud-based strategies, and machine learning algorithms to self-manage Parkinson's disease. Melillo and Scala created a cloud-based mobile health platform for hypertensive patients that allows for remote monitoring and clinical decision-making in a similar manner. To assist patients, Ahsan et al. created a SkinGuard app that analyses their smoking habits using machine learning algorithms and communicates with their cloud-based data storage [2] [5]. These based on the cloud techniques relocate inference execution there, however, this has several disadvantages, such as:

- **Latency:** Latency is a unit of metric used to describe how quickly a service that uses the cloud responds to a customer's inquiry [1].
- **Privacy**: When using cloud computing for medical and health applications, transferring important information there may result in privacy problems.
- **Price**: Cloud-based techniques are paid for by cloud service providers [3].
- **Connectivity**: Although cloud-based services might not always be accessible, running cloud-based programs requires a network connection.
- **Customization:** Typically, standard designs built around communal databases are offered by services in the cloud.

- These simulations could prove to be appropriate or adaptive for certain health problems [6].

2. Application Design and Methodology

2.1 On-device Inference

Training and inference are the two stages of supervised machine learning. Machine learning on a mobile device faces a hurdle because training is frequently computationally intensive and takes days or even months. In order to address this issue, this study used a powerful computer for the training phase and a mobile device for the inference phase.

After a model was trained, it was put into use on a mobile device. All of the inference calculations are conducted locally by our app.

The main benefit of performing inference directly on a mobile device is efficiency. The App can automatically classify fresh data because it doesn't need to communicate with the cloud server [7] [8]. Furthermore, patients are not required to transfer their data, including any Protected Health Information (PHI), to the cloud server, removing any chance of a privacy breach.

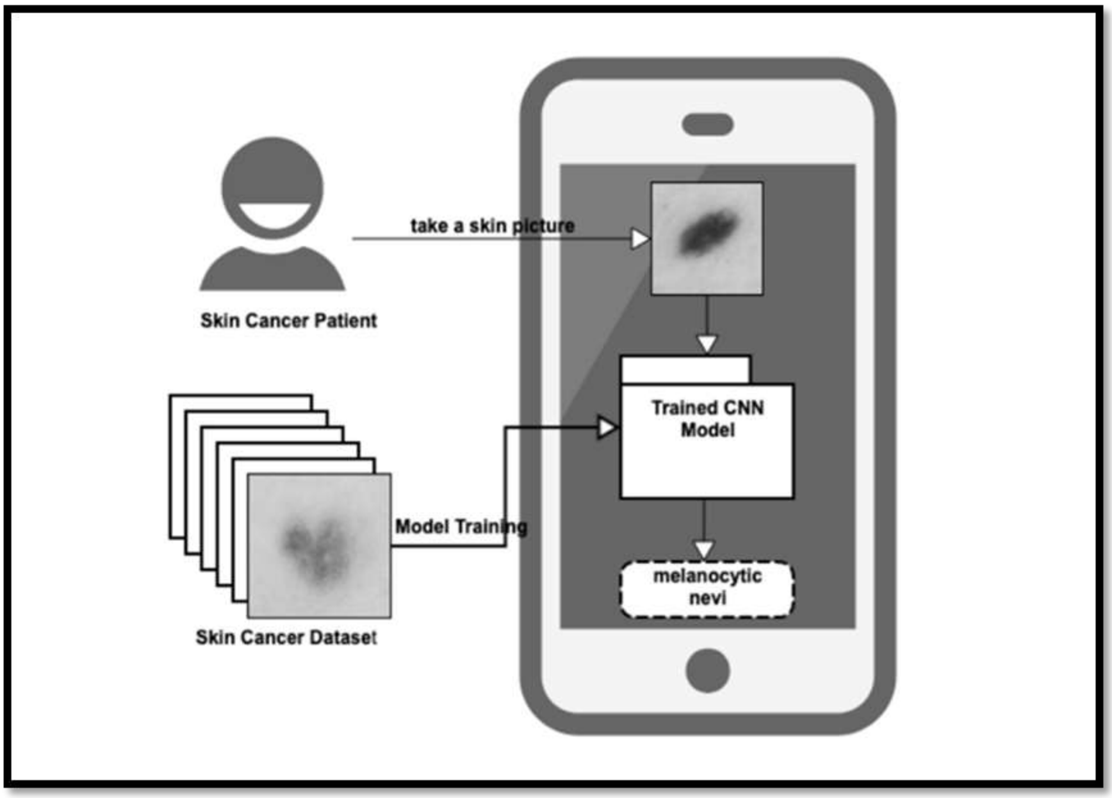


Fig. 1 - On-device Inference App for Skin Cancer Detection

Images of skin lesions that have been classified according to their type are included in the training dataset for our case study. The classification model has been developed to make distinctions between several types of skin tumors [8]. The model is used to classify a fresh skin picture that may have been taken using a mobile device according to its kind during the inference phase.

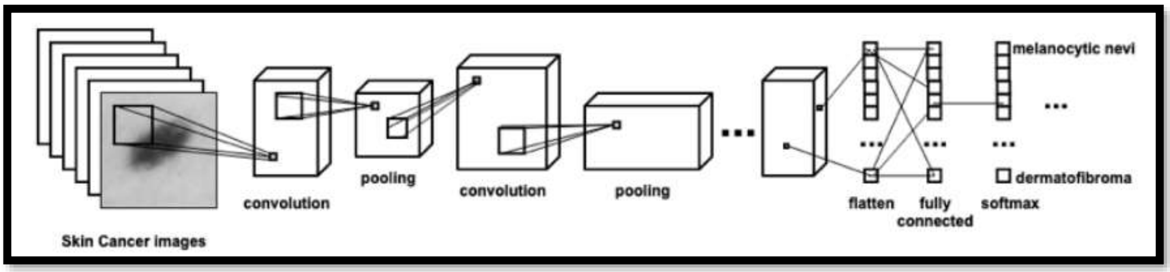


Fig. 2 - Architecture of Convolutional Neural Networks for Skin Cancer Detection

2.2 Convolutional Neural Networks

A particular kind of neural network known as "convolutional neural networks" (CNNs) connects two mathematical algorithms to produce a third by the use of at least one circuit layer. The most typical application for them is picture classification [9]. LeCun et al. introduced the idea of CNNs early on by using just one convolution layer. Since then, academics have continuously enhanced the initial approach, as seen in the development of AlexNet. A CNN's key advantage is its ability to automatically extract relevant information. In our dataset, skin photos are split into seven categories, and the CNN autonomously learns the characteristics of each category.

3. Experiments and Implementation

3.1 Dataset

To provide proof of concept, we use a collection of malignant melanoma photos [2]. There are 10,015 dermatoscopic images in the collection, which are divided into seven different categories of skin lesions: basal cell carcinoma, benign keratosis-like lesions, melanocytic nevi, melanoma, actinic keratosis, vascular lesions, and dermatofibroma [6] [7] [10].

Table 1 - An example of a table.

Skin lesion type	**Abbreviation**	**Total**
Melanocytic nevi	nv	6705
melanoma	mel	1113
Benign keratosis	bkl	1099
Basal cell carcinoma	bcc	514
Actinic keratosis	akiec	327
Vascular lesions	vasc	142
dermatofibroma	df	115

Each image has a record in the metadata table, which includes fields such as lesion identification, image identifier, diagnosis, diagnosis type, age, sex, and localization. Table 2 contains instances from the metadata table. After inspecting the dataset, we detected 57 records with null or missing values and deleted them from consideration. The remaining photos were then scaled to 120*90 for practical reasons because the original size, 600*450, was too huge to train in TensorFlow [2].

3.2 Model Training

The fundamental algorithms were developed on macOS using Python, Scikit-learn, TensorFlow, and Keras. Several popular machine learning algorithms are included in the Python module Scikit-learn. TensorFlow is an open-source deep-learning model library. To facilitate effective experimentation, Keras runs on top of TensorFlow. We employed the architecture depicted in Figure 2, which employs convolutional land pooling layers. Pooling is a non-linear down-sampling commonly used after convolutional layers to remove extraneous features and limit the number of parameters during training to avoid overfitting. To be more precise, we employ max pooling (MaxPool2D). Following the convolutional and max pooling layers, we employ the flattening and fully connected (FC) layers to flatten the multidimensional array into a two-dimensional one [11]. The output layer is built on the soft-max function, which computes the probabilities of each class (i.e., type of skin lesion) to recommend the most likely classification. The soft-max function is depicted mathematically below:

$$\sigma(Z)_j = \frac{e^{Z_j}}{\sum_{k=1}^{K} e^{Z_k}} \quad \text{for} \quad j = 1, \ldots, \mathrm{K} \tag{1}$$

Where z is a vector of the inputs to the output layer and j indexes the output units.

One of the difficulties in machine processing skin images involves the fact that they may be obtained under a variety of conditions (e.g., brightness, angle, focal distance), making image comparison and identification of key parts difficult. Data augmentation was used, which in this case uses existing photographs that have been randomly rotated, zoomed, moved, and cropped from the center to produce new images in order to lessen the influence of image properties on the classification model [12]. Then, you can utilize these additional pictures to enhance your education. we divided the dataset into three sections for training (64%), validation (16%), and testing (20%), accordingly. During the training phase, the validation data were used to fine-tune the hyperparameters [13]. The best outcomes that we attained are shown in Table 3

Table 2 – Samples of Metadata.

Lesion Identifier	Image Identifier	Diagnosis	Diagnosis Type	Age	Sex	Localization
HAM_0001480	ISOC_0026835	bkl	histo	70	M	abdomen
HAM_0005388	ISIC_0027815	Bkl	histo	80	M	chest
HAM_0002129	ISIC_0025903	df	consensus	60	M	abdomen
…	…	…	…	…	…	…
HAM_0004607	ISIC_0026150	mcl	histo	50	F	back
HAM_0006092	ISIC_0029241	mcl	histo	70	M	face
HAM_0006047	ISIC_0026321	bcc	histo	65	F	scalp

Applying CNN to the provided dataset yields the pre-trained model. The model we trained is in HDF5, a file format suitable for storing massive collections of multidimensional numeric arrays (e.g., models, and data files) [13].

Table 3 – Model Evaluation.

Data Augmentation				Performance	
Rotation Degree Range	**Zoom Range**	**Width Shift Range**	**Height Shift Range**	**Loss**	**Accuracy**
10	0.1	0.15	0.15	0.71	75.2%

Converting models and incorporating pre-trained models into the app

Android's Core ML machine-learning framework (Figure 3) allows us to embed learned machine-learning models into Android apps [16]. The integration procedure transfers the cloud-based machine learning model to mobile devices.

Core ML, however, only supports a tiny fraction of model types. In most cases, we need to convert the pre-trained model into one that can be integrated into the mobile app. In this study, we used the Python module Core ML Tools to transform our pre-trained model (.hdf5 format) into the Core ML model format [11] [12].

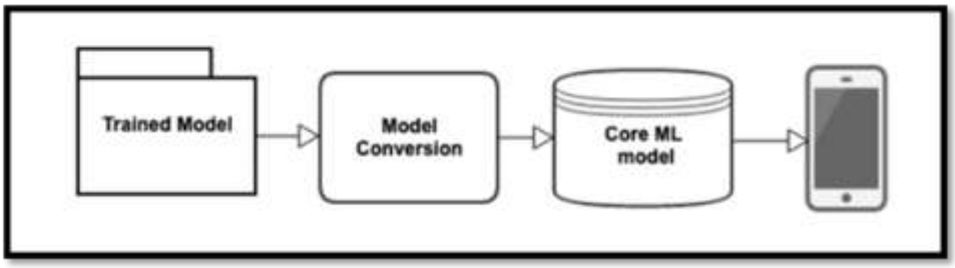
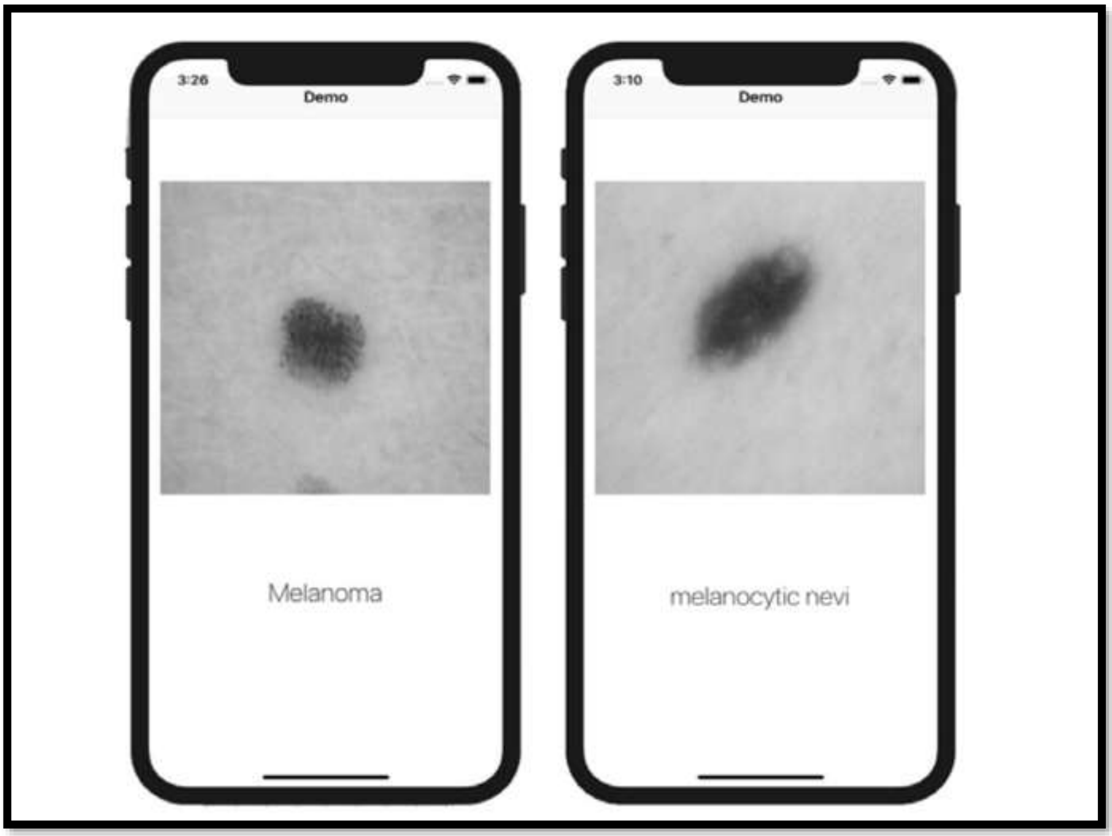


Fig.3 - Model Conversion and Integrating Pre-trained Model into the App

3.3 Demonstration

The app's UI, created for the Android operating system, is seen in Figure 4. A skin image is supplied by a user [15]. Then, a probability distribution over the categories of skin lesions under consideration is created using this image. The most likely classification (for example, melanoma) is subsequently presented on the screen. The inference procedure is almost instantaneous.

Privacy concerns must be addressed in the context of medical applications. In this paper, we offer an on-device inference strategy with several advantages over cloud-based systems. They include decreased latency, increased availability, enhanced privacy, and decreased cost. It is not essential to activate a cloud service in order to classify the type of skin lesions because the pre-trained model is already deployed on the mobile device [5] [6] [7].



This on-device inference method significantly lowers latency. By not requiring patients to send photographs to a third-party online server, it additionally enhances security. Last, but not least, it eliminates the overhead and costs related to operating and maintaining cloud services.

4. Conclusion

Skin cancer is a deadly disease that affects millions of people worldwide. Early detection is critical for successful treatment and the prevention of the spread of the disease. Machine learning algorithms have shown considerable promise in diagnosing skin cancer through picture analysis. The application of deep learning represents one of the most common computational learning algorithms used to identify cancer of the skin with regard to convolutional neural networks (CNNs). An artificial neural network called a CNN has demonstrated extraordinary performance in image classification and segmentation tasks. training CNNs on large datasets of skin disease photos, they can learn to accurately distinguish between benign and malignant lesions.

To build a skin cancer detection system, researchers typically start by collecting a large dataset of skin lesion images. This dataset can consist of images from various sources, including medical databases and publicly available datasets. The images are then labeled by trained dermatologists to identify whether the lesion is benign or malignant [14].

Once the dataset is prepared, the researchers can start building the machine-learning model [13]. The model is typically built using a pre-trained CNN architecture, such as VGG-16 or ResNet that has been trained on a large dataset of general images. The pre-trained model is then fine-tuned on the skin lesion dataset by adjusting the weights of the CNN's layers to better suit the task at hand.

The performance of the skin cancer detection model is evaluated by using metrics such as accuracy, sensitivity, and specificity. Accuracy measures the overall percentage of correct classifications made by the model, while sensitivity measures the percentage of malignant lesions that were correctly identified as malignant. Specificity measures the percentage of benign lesions that were correctly identified as benign. Recent studies have shown that machine learning models can achieve high levels of accuracy in skin cancer detection. A journal of the American Medical Organization, for example, published an investigation in 2018 that showed that a CNN-based model was able to achieve a classification accuracy of 90.3% in distinguishing between benign and malignant skin lesions. Another study published in the Journal of Investigative Dermatology in 2019 showed that a deep learning algorithm was able to achieve a sensitivity of 90.3% and specificity of 87.2% in detecting melanoma [14]. While machine learning algorithms have shown considerable promise in skin cancer detection, several problems remain. One of the major challenges is the lack of diverse and representative datasets, particularly for underrepresented skin types. Additionally, there is a need for robust and interpretable models that can explain the decisions made by the algorithm. In conclusion, skin cancer detection using machine learning is a fascinating area of research with the potential to enhance the early diagnosis and treatment of this fatal illness [15]. By leveraging the power of deep learning algorithms, researchers can accurately distinguish between benign and malignant skin lesions and help save lives. However, more work needs to be done to address the challenges associated with this technology and ensure that it is accessible to all populations.

REFERENCES

[1] Khan, Sameer, and Suet-Peng Yong. "A deep learning architecture for classifying medical images of anatomy object." Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017. IEEE, 2017.

[2] Tschandl, Philipp, Cliff Rosendahl, and Harald Kittler. "The HAM10000 Dataset: A Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions." arXiv preprint arXiv:1803.10417 (2018).

[3] Armstrong, April W., et al. "Text-message reminders to improve sunscreen use: a randomized, controlled trial using electronic monitoring." Archives of Dermatology 145.11 (2009): 1230-1236.

[4] Youl, Philippa H., et al. "Can skin cancer prevention and early detection be improved via mobile phone text messaging? A randomized, attention control trial." Preventive medicine 71 (2015): 50-56.

[5] Banos, Oresti, et al. "mHealthDroid: a novel framework for agile development of mobile health applications." International Workshop on Ambient Assisted Living. Springer, Cham, 2014.

[6] Gatsios, Dimitrios, et al. "Mhealth platform for Parkinson’s disease management." International Conference on Biomedical and Health Informatics. Springer, Singapore, 2015.

[7] Melillo, Paolo, et al. "Cloud-based remote processing and data-mining platform for automatic risk assessment in hypertensive patients." International Workshop on Ambient Assisted Living. Springer, Cham, 2014.

[8] Pan, Di, et al. "A mobile cloud-based Parkinson’s disease assessment system for home-based monitoring." JMIR mHealth and uHealth 3.1 (2015).
[14] Callejas, Zoraida, et al. "A virtual coach for active ageing based on sentient computing and m-health." International Workshop on Ambient Assisted Living. Springer, Cham, 2014.

[9] Shin, Hoo-Chang, et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics, and transfer learning." IEEE Transactions on medical imaging 35.5 (2016): 1285-1298.

[10] Kamnitsas, Konstantinos, et al. "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation." Medical image analysis 36 (2017): 61-78.

[11] Tajbakhsh, Nima, et al. "Convolutional neural networks for medical image analysis: Full training or fine-tuning?." IEEE Transactions on medical imaging 35.5 (2016): 1299-1312.

[12] Nie, Dong, et al. "Estimating CT image from MRI data using 3D fully convolutional networks." Deep Learning and Data Labeling for Medical Applications. Springer, Cham, 2016. 170-178.

[13] Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[14] Moeskops, Pim, et al. "Automatic segmentation of MR brain images with a convolutional neural network." IEEE Transactions on medical imaging 35.5 (2016): 1252-1261.

[15] Bar, Yaniv, et al. "Deep learning with non-medical training used for chest pathology identification." Medical Imaging 2015: Computer-Aided Diagnosis. Vol. 9414. International Society for Optics and Photonics, 2015.

[16] Marcolino, Milena Soriano, et al. "The impact of mHealth interventions: a systematic review of systematic reviews." JMIR mHealth and uHealth 6.1 (2018).

[17] Dai, Xiangfeng, and Marwan Bikdash. "Trend analysis of fragmented time series for mHealth apps: Hypothesis testing based adaptive spline filtering method with importance weighting." IEEE Access 5 (2017): 27767-27776.

[18] Longacre, Meghan, et al. "Clinical Adoption of mHealth Technology to Support Pediatric Cystic Fibrosis Care in Sweden: Qualitative Case Study." JMIR Pediatrics and Parenting 1.2 (2018): e11080.

[19] Ahsan, GM Tanimul, et al. "Toward a mHealth intervention for smoking cessation." 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops. IEEE, 2013.