



Design and implementation of a deep learning-empowered m-Health application

Akhan Akbulut¹ · Sara Desouki¹ · Sara AbdelKhalik¹ · Layal Khantomani¹ · Cagatay Catal² 

Received: 23 December 2022 / Revised: 17 August 2023 / Accepted: 11 September 2023
© The Author(s) 2023

Abstract

Many people are unaware of the severity of melanoma disease even though such a disease can be fatal if not treated early. This research aims to facilitate the diagnosis of melanoma disease in people using a mobile health application because some people do not prefer to visit a dermatologist due to several concerns such as feeling uncomfortable by exposing their bodies. As such, a skincare application was developed so that a user can easily analyze a mole at any part of the body and get the diagnosis results quickly. In the first phase, the corresponding image is extracted and sent to a web service. Later, the web service classifies using the pre-trained model built based on a deep learning algorithm. The final phase displays the confidence rates on the mobile application. The proposed model utilizes the Convolutional Neural Network and provides 84% accuracy and 72% precision. The results demonstrate that the proposed model and the corresponding mobile application provide remarkable results for addressing the specified health problem.

Keywords Melanoma · Machine learning · Deep learning · m-Health · Skin lesion analysis

1 Introduction

Melanoma is an extremely common form of skin cancer, which is caused by the abnormal growth of melanocytes (i.e., the cells that give the skin its tan or brown colour). When cells go out of control, cancer develops and spreads to many parts of the body. Unfortunately, melanoma can be the deadliest of all skin cancer diseases; however, it can also be treated if diagnosed early. Even though the sun has a significant role in causing melanoma, it is found that Ultraviolet (UV) light exposure can be responsible for most melanomas (i.e., 86% of melanomas are caused by UV). One can also get such a disease without being exposed to the sun since melanoma can be genetically inherited. Speaking of genes, melanoma affects

✉ Cagatay Catal
ccatal@qu.edu.qa

¹ Department of Computer Engineering, Istanbul Kültür University, 34536 Istanbul, Turkey

² Department of Computer Science and Engineering, Qatar University, Doha 2713, Qatar

men and women differently. Melanoma is more frequent in women than in men under the age of 49. Men are more likely than women to die of skin cancer by the age of 50. This may be because melanoma frequently occurs on parts of the body that are more difficult to track in men. It is more prevalent in men than in women after the age of 50.

There is not only one type of melanoma but three types of Melanoma are common. Nodular melanoma can have different colours that are pink, black, or red. They can also appear as clear bumps. Lentigo malignant melanoma is usually found in older people. Its shape is like a flat light brown smear that expands and darkens with time. Acral lentiginous is another type of melanoma that appears as black bands on the wrist's palm, feet's soles, and nails. New technologies were able to diagnose skin cancer by using machine learning algorithms. In this research, we aim to develop a mobile application using Artificial Intelligence (AI), particularly Machine Learning (ML), to track skin spots. This application is supported by an ML model that can classify skin images as malignant or benign.

The novelty of this study is mainly related to the proposed system architecture. The system's integration of various modules, including the mobile application, web server, database, classification system, and cloud storage, demonstrates a seamless and efficient flow of data. The novel architecture allows for quick and accurate diagnostic results for users. The adoption of specific technologies, such as Flutter for the client application, NodeJS for the server, MongoDB for the database, and Flask for deploying the machine learning model, showcases a comprehensive and contemporary approach to building the system. While some preliminary research that uses CNN algorithms has been carried out in the literature, they have not considered the use of state-of-the-art technologies as used in this research. Furthermore, the creation of a web service acting as a bridge between the mobile application, MongoDB database, and Google Cloud Storage buckets demonstrates our system's ability to connect and interact with multiple data sources, making it a versatile and robust solution. Finally, while other methods of diagnosing melanoma exist, our research stands out for introducing a user-friendly and accessible solution through a mobile application.

The contributions of this research are three-fold:

1. A user-friendly mobile application that diagnoses skin spots with confidence rates was developed.
2. A machine learning model to classify melanoma images as malignant or benign was built.
3. A web service to connect the mobile application with the machine learning model was implemented in this research.

2 Related works

Research confirms that regular access to health care is an effective strategy to prevent the causes of poor health. Biomedical engineers state that early diagnosis can help save more than 40% of the lives lost [1]. According to industrial research, the Google Play Store and Apple iTunes have over 165,000 health-related smartphone applications (a.k.a., apps) [2]. Nowadays, medical apps are developed using artificial intelligence and, particularly, machine learning techniques to aid in diagnosis. Researchers can build algorithms and models to identify objects, whether it is a cat or dog, however, in medicine, we rarely have ground truth values [3]. The best approach to getting confidence in models is to put them through thorough validation and testing stages

Melanoma classification using machine learning, deep learning, and neural networks has received a lot of attention in the past few years. This is because AI has progressed dramatically; deep learning-based algorithms have been developed that can automatically extract features and learn from them. Gupta et al. cited many articles related to CNN and compared them based on the classification technique they used and the accuracy outcome. The best accuracy achieved was 98.7% using the CNN and a dataset of clinical images.

TensorFlow and Keras are helpful software libraries to develop a powerful deep learning model. As for the dataset, it is divided into several sets (i.e., training, validation, testing). Benbrahim et al. [5] split their dataset into the following sets: 80% training, 10% validation, and 10% testing sets. Their CNN architecture made use of a convolutional layer and then a max-pooling layer, in which this pattern was repeated four times. Also, they added a flattening layer, a dropout layer, and ended the model with two dense layers. Their proposed model achieved 93.93% in the testing set.

Another study that uses CNN architecture is by Namozov and Cho [6]. The dataset they used contained 10,015 images and was taken from ISIC 2018. The CNN was used as the classification method, with a total of nine layers. The network has four convolutional & two pooling layers for feature extraction and three fully connected layers at the end for the classification part. The special part about this CNN is that a piecewise linear activation function was used in the convolutional layers, which enhances the performance of the network. The first experiment was based on the ReLU function, which resulted in 93.25%. The tangent function provided the result of 91.76%. Lastly, the piecewise activation function returned the best result of 95.86%. With more training sets, the CNN using the piecewise function reached 98% accuracy after 65 epochs.

Another study that used the same dataset (ISIC 2018) was done by Guha et al. [7]. The first step is the image preprocessing part, which focuses on enhancing the images' qualities. Then, the images are taken as inputs and processed with a median filter to remove the image noise. Moreover, feature extraction takes part where the Otsu thresholding technique is used to decrease the grey-level image. The last step of the pre-processing step is the classification, where CNN was applied. The CNN used two max-pooling, and 2 fully connected layers. In the first layer, 32 convolution channels were chosen with a size of 3x3 developing 32 channels of 396 x 644. The second layer contained 64 convolution layers, which use 64 channels of 394 x 642. Two convolution layers on the input images were added, and the max-pooling was set to 2x2. In the third layer, 64 convolution channels of size 195 x 319 were utilized. The stride of 2x2 and kernel size of 2x2 were utilized in the max-pooling layer. Dropouts were linked with SoftMax, and the model was trained for 40 epochs. The provided result of the model was 79.42%.

A deconvolutional neural network (DDN) based on residual learning for skin lesion segmentation can help address the wide range of skin lesions [8]. Dense deconvolutional layers (DDLs), chained residual pooling (CRP), and hierarchical supervision (HS) are all parts of the proposed network. DDL allows the proposed DDNs to reuse learned features from previous layers, forming tight linkages between all feature maps. Another study [9] uses a convolutional-deconvolutional neural network (CDNN) that helps in tumour segmentation. The dataset is from ISBI 2017 for both training and validation. All images were re-sized to 192 x 256 pixels by using the bi-linear interpolation. Additionally, HSV was used instead of RGB to apply image augmentation. The kernels applied for this CDNN are of size 3x3 in both the convolutional and de-convolutional layers. Lastly, batch normalization is applied, and the batch size is 18. The CDNN contained 29 layers with 5,042,589 trainable parameters. The experiment provided a testing accuracy of 93%.

ResNet is a classification method that helps in feature extraction. A study [10] suggests an in-depth study framework for end-to-end skin analysis. The proposed framework deals with multiple tasks simultaneously and uses ResNet50 (50 layers) and ResNet101 (101 layers) to extract melanoma features. It also uses an active loss function to reduce the problem of inequality in the skin data section and utilizes a region proposal network to remove background information. It provided the best results using segmentation-with-different-hyperparameters, IMAGENET Initialization, and a batch size of 2X2. During the testing stage, the accuracy of ResNet50 got 97% while ResNet101 provided 96% accuracy. In another study, ResNet 50, 40, 25, 10, and 7 models were used and compared to reach the best accuracy. According to the researchers [11], the best model in this study was Resnet50 using 10-epoch training with data augmentation.

Gonzalez-Diaz [12] used both CNN and Resnet-50 for training the model, and also used DermaKNet which is a CAD system for the automatic diagnosis of skin lesions. The dataset used in this study is the ISBI, which contains 2750 dermoscopic images with varying resolutions. The dataset was split into three parts which are training (i.e., 2000 images), validation (i.e., 150 images), and testing (i.e., 600 images). All of these images were then adjusted to 256 x 256 pixels. The Area Under ROC Curve (AUC) was 87.4% for Resnet-50 and 95% for the CNN-based model.

Google also developed a classification method called MobileNetV2, which is based on the CNN architecture, specifically designed for mobile and embedded applications. Akay et al. [13], stated that MobileNetV2 showed better results than the traditional CNN algorithm. The suggested network achieved 94.8% on the testing dataset.

Region-of-interest (ROI) can be extracted using other classification methods. Ashraf et al. [14] identify and differentiate melanoma from nevus cancer using a CNN-based approach. The system uses enhanced K-Means to extract ROIs from images and is trained and tested on DermIS and DermQuest images. This framework achieved the best accuracy with data augmentation and an ROI-based approach, 97.9% on the DermIS dataset and 97.4% on the DermQuest dataset.

Deep convolutional neural networks can be applied on mobile phones using a web service, TFLite, or TensorFlow Mobile. Emuoyibofarhe et al. [15] performed a comparison between TensorFlow Mobile and TFLite to find which one suited better. The results state that TFLite is a solution for making machine learning models on mobile and embedded devices. However, it does not support all TensorFlow operations. Tensorflow mobile, on the other hand, supports more features. In this research, our mobile app uses web service technology.

Table 1 shows the comparison of different research studies discussed in this study. In a recent review article [19], researchers presented a review of state-of-the-art machine learning techniques used to detect skin cancer. From this review paper, we observed that although there are various algorithmic-level solutions, there is a lack of a comprehensive and contemporary approach that explains how to build a classification system using state-of-the-practice software tools. In this article, we present how to design and implement such a system using current tools and technologies.

3 Methodology

In this section, we introduce the proposed melanoma detection system. Section 3.1 provides the system architecture. Section 3.2 is about the tools and technologies used. In Section 3.3 section, the proposed neural network-based model is explained. In Section 3.4, the web

Table 1 Related works comparison table

Reference No. #	Publish Year	Dataset	Method	Metrics
Gupta et al. [4]	2020	Best Using Clinical Images	CNN	Accuracy=99%, Sensitivity=95%
Benbrahim et al. [5]	2020	HAM100000	CNN	Validation accuracy=94%, Testing accuracy=94%
Namozov and Cho [6]	2018	ISBI 2018	CNN	Accuracy: 95.86%
Guha and Rafizul Haque [7]	2019	ISIC 2018	CNN Extraction: Otsu Thresholding	Accuracy: 79.42%
Li et al. [8]	2018	ISBI 2016 and 2017	Classifier: CNN based on Deconvolutional neural network	(ISBI 2016) Accuracy=96%, Dice coefficient=93%, Jaccard Index=87%, Sensitivity=95% Specificity=96% (ISBI 2017) Accuracy=83%, Dice coefficient=87%, Jaccard Index=77%, Sensitivity=83% Specificity=98%
Yuan and Lo [9]	2019	ISBI 2017	CDNN Segmentation: bi-linear interpolation	Accuracy=95%, Dice coefficient=87% Jaccard Index=78%, Sensitivity = 88% Specificity=98%
Song et al. [10]	2020	ISBI 2016 and ISIC 2017	ResNet50, ResNet101 Segmentation: Loss functions: Dice Loss, Focal Loss, Jaccard loss, FL+JL, FL+JL+DL	(ResNet50) Test Accuracy = 97%, Dice coefficient = 91% Jaccard Index = 84%, Sensitivity = 86% Specificity = 99% (ResNet101) Test Accuracy = 96%, Dice coefficient = 90% Jaccard Index = 83%, Sensitivity = 86% Specificity = 98%
Budhiman et al. [11]	2019	ISIC 2018	Resnet5	Testing accuracy: 83% f1 (malignant class)=43% f1 (benign class)=87%
González-Díaz [12]	2019	ISBI 2016	CNN and Resnet-50 Segmentation: DSSN	CNN's Accuracy: 95% and Resnet-50's Accuracy: 87.4%
Akay et al. [13]	2021	ImageNet dataset	MobileNetV2	Training accuracy=100% Validation accuracy=97.2% Testing accuracy=94.8% Sensitivity=95%
Ashraf et al. [14]	2020	DermIS and DermQuest, best with data augmentation	CNN Extraction: K-Means	(DermIS) Accuracy=98%, Recall= 98%, F1=98% (DermQuest) Accuracy=97%, Recall= 98%, F1=98%

Table 1 continued

Reference No. #	Publish Year	Dataset	Method	Metrics
Castro et al. [15]	2020	ISIC with data augmentation	Google Inception V3	Training accuracy=90%, Test accuracy=81%, ROC=88%, Sensitivity=84%, Precision=79%
Our Study	2022	ISIC with data augmentation	CNN	Training accuracy=99%, Validation accuracy=99% Testing accuracy=84%, Precision=72% Recall=40%, F1=51% ROC_AUC=64%

service is discussed, and in Section 3.5, the implementation details of the mobile application are presented.

3.1 System architecture

The proposed system consists of the following five modules: The mobile application, web server, database, classification system hosted on a server, and cloud storage. Patients interact with the mobile application to view their logs and their profiles. When a patient adds a new entry to their log lists or retrieves old logs, the mobile application communicates with the web server, which in turn performs CRUD operations on the database to send data back to the mobile application. When a patient captures a new photo for diagnosis purposes, the mobile application communicates directly with the web server, which sends back the diagnosis result. To save the log photos once they are captured, Google Cloud Storage is used. The application was developed in accordance with the microservice architecture [16]. The order of communication between the different parts of the system is shown in Fig. 1.

3.2 Tools and technologies

3.2.1 Dataset

The dataset had a total of 53,081 images, which were later split into training and validation sets. The training dataset takes 80% of the images, and the rest is used for validation. The size of the images is 224x224 and was accessed from multiple sources on Kaggle [17] and *SensioAI* [18]. As for the testing part, there was an external dataset used, which contains 4,100 images.

3.2.2 Tools

The Jupyter Notebook was used to train the models, and a GPU environment was created. Flutter was used to build the client application, NodeJS was utilized for building the server for the application, and MongoDB was used for creating the database.

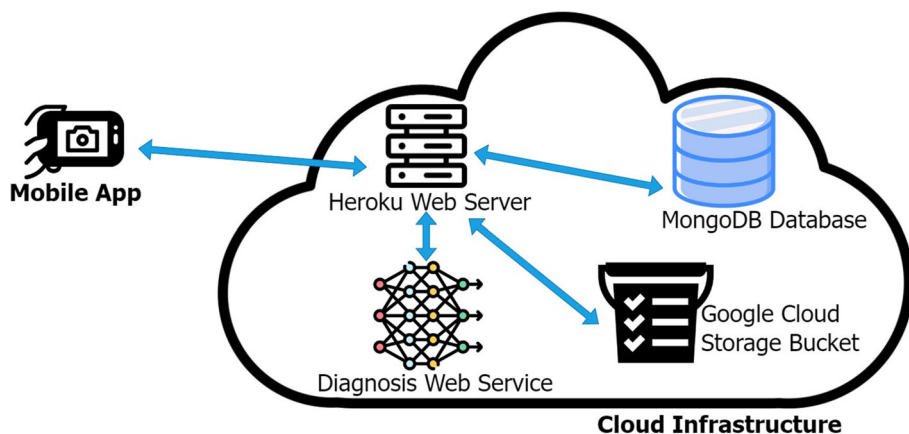


Fig. 1 System Architecture diagram

3.3 The proposed model

The input of our neural network is of size 224x224x3. Our first layer is Conv2d, which is the feature extractor layer. Then, we have a max-pooling layer that helps to extract low-level features. After that, we used dropout, which is a technique that ignores a certain rate of neurons to prevent overfitting. The dropout rate was 0.25. Then, we repeated this pattern, and the model had in total 2 convolutional layers, 2 max-pooling layers, and 2 dropout layers. Later, we flattened the output into a one-dimensional array and created a fully-connected layer. The output layer determines whether the mole is benign or malignant, as represented in Fig. 2. We started with 32 neurons, then 64, and finally, the fully connected layer was composed of 128 neurons.

We used the normal initialization approach that samples the weights in a normal distribution with a low deviation. As for the activation function, we used Relu and for the optimizer, adam was preferred.

3.4 Web service

In order to deploy the machine learning model, we used the Flask web framework and our model as a file with the '.h5' extension. We used the POST method to get the image from the

Conv2d__1: Conv2D	input	(None, 224, 224, 32)
	output	(None, 224, 224, 32)
Max_pooling2d_1: Max_pooling2d	input	(None, 224, 224, 32)
	output	(None, 112, 112, 32)
Dropout_1: Dropout	input	(None, 112, 112, 32)
	output	(None, 112, 112, 32)
Conv2d__2: Conv2D	input	(None, 112, 112, 32)
	output	(None, 112, 112, 64)
Max_pooling2d_2: Max_pooling2d	input	(None, 112, 112, 64)
	output	(None, 56, 56, 64)
Dropout_2: Dropout	input	(None, 56, 64, 64)
	output	(None, 56, 56, 64)
Flatten_1: Flatten	input	(None, 56, 56, 64)
	output	(None, 200704)
dense_1: Dense	input	(None, 200704)
	output	(None, 128)
dense_2: Dense	input	(None, 128)
	output	(None, 2)

Fig. 2 Architecture of CNN Model

gallery. Then, we decoded that image and opened it by wrapping it into IO. BytesIO because it is not an actual file yet. We pre-processed the taken image from the gallery and converted it to RGB. The target size of our image was 224 X 224. After that, we turned the image into an array and expanded that array by inserting a new axis at position 0. Eventually, we decided that the image based on the model where 0 represents the benign case and 1 shows the malignant case, and this was returned as a JSON message to the front end.

3.5 Mobile application

3.5.1 Client app

Flutter framework was used to develop the app. The Consumer-Provider approach was used to manage the state of the application. A models directory is used to keep track of the class schemas, containing a separate file for each schema and its functions. For navigation and routing between pages, the named routes method is used. A routes directory is used to keep the list of route names, and the custom router class is responsible for managing navigation using the given route names. The screens directory contains a separate file for each screen used in the application, containing its layout and its respective functions. The services folder contains services that have methods that are responsible for communicating with the diagnosis web service or the mobile server. The HTTP protocol is used for communicating with web services with the help of a library called Dio that makes sending and receiving HTTP requests easier than using plain HTTP.

3.5.2 Server

A web service was created to act as a bridge between the Flutter application and both the MongoDB database and the Google Cloud storage buckets. The responsibilities of the web service can be summarized into the following two main categories:

- Connecting the application with the MongoDB database and performing all CRUD operations on the database. This refers to all operations that can be performed in the LogList screen in the application (i.e., adding entries, deleting entries, editing folders, retrieving the log list, and saving all mole images taken by the user to be easily accessible even if the user uninstalls the application).
- Managing the Login and SignUp operations. This includes encoding user passwords at registration, storing and handling user information in the database, and authenticating registered users.

To create this web service, NodeJS was used, specifically the ExpressJS framework. The Passport library was used for the authentication process (i.e., user login and sign-up). Bcrypt was used for encrypting and decrypting user passwords when creating new accounts or during authentication when logging in. To communicate with the database, JSON was used.

Full documentation for the methods and the classes of the server was generated using JSDoc markup language. A library called Swagger was used to create interactive documentation for the HTTP requests, documenting all necessary headers, queries, and parameters needed in all different requests performed across the application's life cycle. The code of the server is divided into three main directories (or modules, in expressJS terminology).

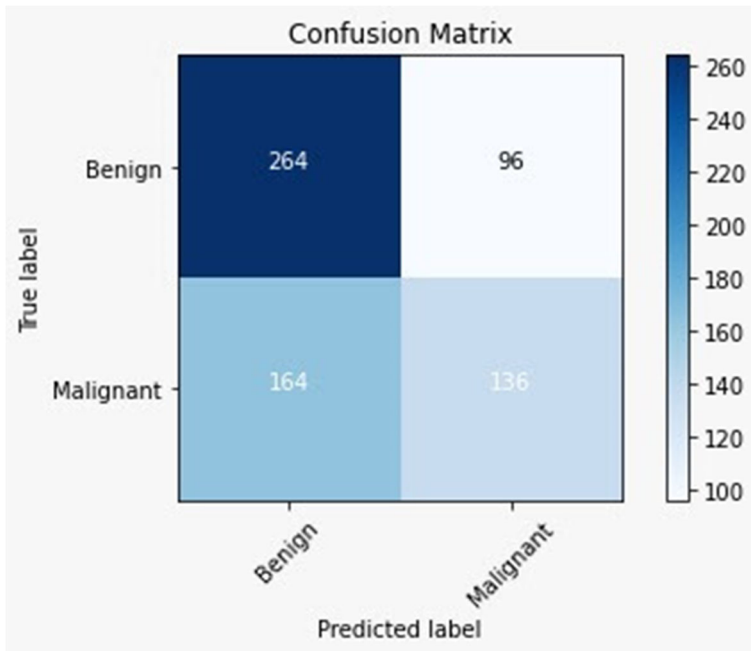


Fig. 3 Confusion Matrix before Data Augmentation

Train ACC	99%
Valid ACC	99%
Test ACC	84%
Precision	72%
Recall	40%
F1	51%
ROC_AUC	64%

Fig. 4 Metrics after Data Augmentation

The methods directory is where all of the actions performed on the user's profile or log are written. Actions are basically expressJS middleware functions, and their job is to perform CRUD operations on the database using Mongoose as a mediator, and they use JSON for data transfer and for returning feedback on whether or not a certain request has succeeded. The models directory contains the database schema definitions used in the MongoDB database. There are two main schemas; user, which stores all information in a user's profile, and log list, which stores all information related to user moles and mole images. The routes directory is where the HTTP requests are defined, organized, and documented for Swagger.

3.5.3 Database

Mongodb was used, and hosted on Mongodb Atlas cloud service, free tier. We chose to use MongoDB since it is more flexible and easy to use than relational SQL databases.

4 Results

4.1 Without data augmentation

We first started training our data using the images. We used 29,454 images for training and 7,364 images for validation. The result for both training and validation accuracy was 99%.

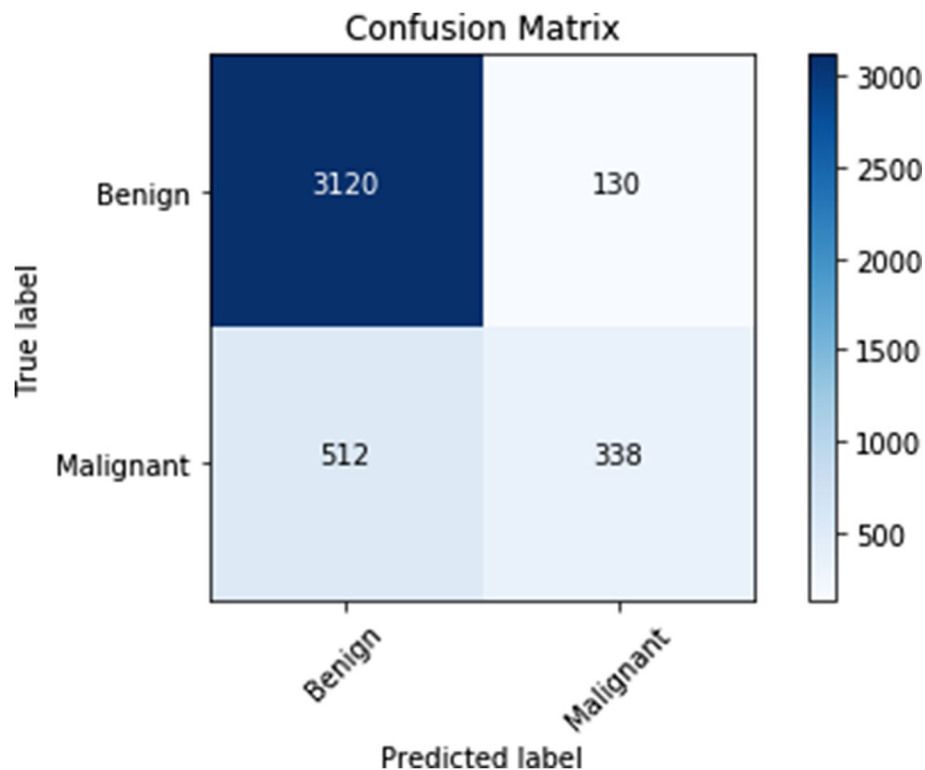


Fig. 5 Confusion Matrix after Data Augmentation

However, when we tried to test our model using 660 unseen images, we got 61% accuracy, as shown in (Fig. 3), which indicated that we had an over-fitting problem.

4.2 With data augmentation

To address the over-fitting problem, we tried to reduce the complexity of the CNN architecture by reducing the learning rate and also applying different regulation methods. Since none of these methods solved our problem, we decided to use the data augmentation approach. As a result, our training images increased to 42,464 and validation images to 10,617. The training and validation accuracy was 99% (Fig. 4). Then, we tested them on 4,100 unseen data and we received 84% accuracy, as shown in Fig. 5, which means that the over-fitting problem was reduced significantly. A precision of 72%, recall of 40%, F1 of 51%, and ROC_AUC score of 64% were reached. The ROC graph is shown in Fig. 6.

4.3 Application

The outcome of the mobile application is shown in different figures. Figure 7 shows the login and main screens of the application. The user can register a new account from the registration screen or log into an existing account from the login screen. After that, they are able to diagnose their moles and save them in their log list, as shown in Fig. 8b, which is where all the diagnosis information is kept in the form of a list of folders containing multiple entries. On the camera screen (Fig. 8a), the user can take a photo by positioning the mole in the focus circle. Later, the photo is cropped and resized to be processed by the ML model.

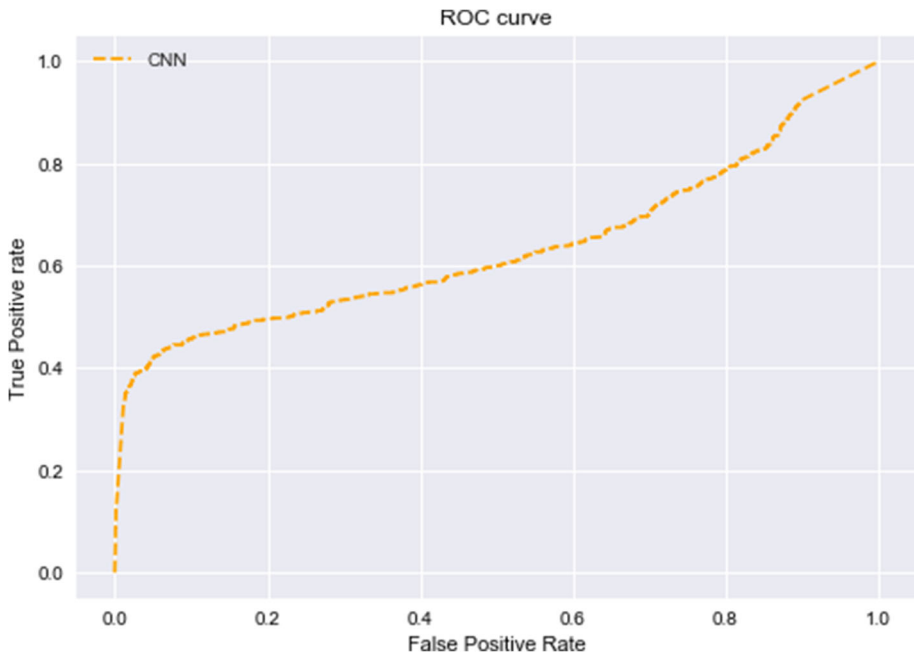


Fig. 6 ROC Curve Graph after Data Augmentation

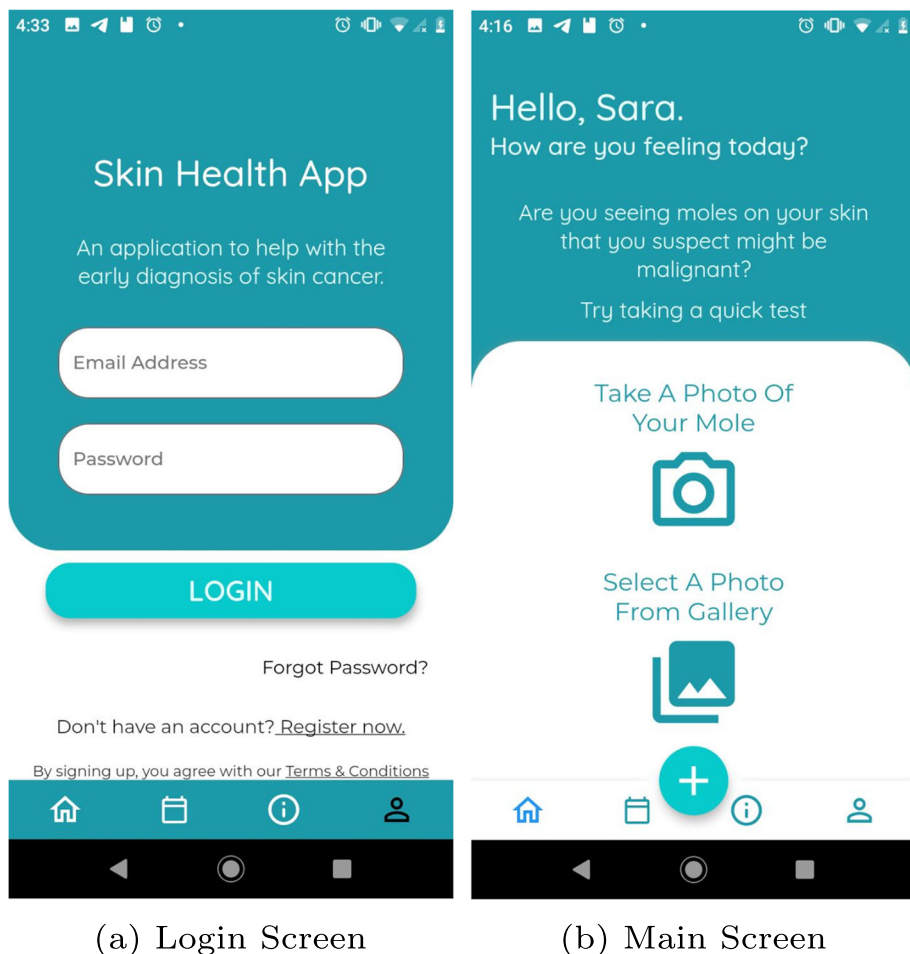


Fig. 7 Login and Main Screens of the Application

Once the user confirms the photo, it is sent for diagnosis. When the diagnosis is ready, the diagnosis report screen appears with information about the classification and the confidence rates matrix. Figure 9 shows some sample screens regarding these classification results. The results are automatically saved to the log list, where the user can view them again later if needed. The current apk size of the application is approximately 66 MB, however, the actual size of the appbundle is yet to be determined if we decide to release the app on Google Play in the future.

5 Conclusion

Finding a very large medical dataset that is precise and correctly formatted is very difficult. Most of the melanoma datasets on Kaggle had only 2% malignant images, while the rest were benign images. Therefore, we had to combine two datasets to address this issue. We

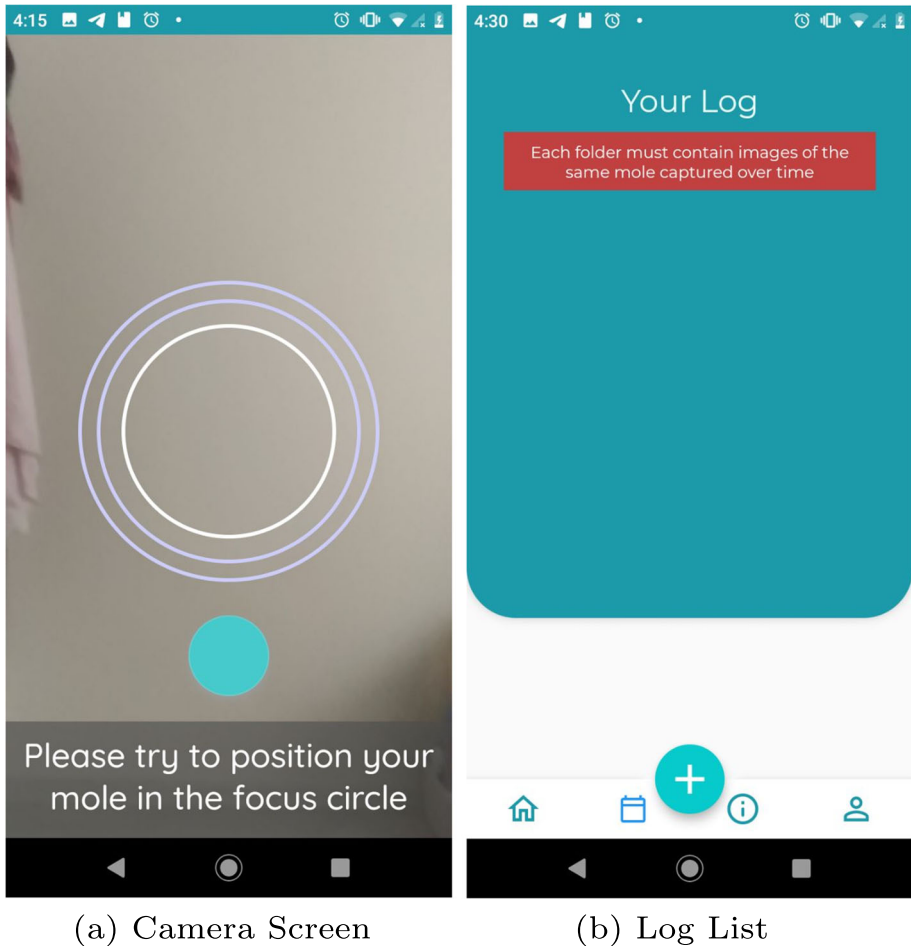


Fig. 8 Camera and Log List Screens

also encountered an overfitting problem and solved using augmented images with higher CPU/GPU. As a result, the training accuracy increased from 84% to 99% and the testing accuracy from 60.6% to 84.3%. As for the mobile application, we connected it with our model successfully and were able to receive a real-time diagnosis. Eventually, we created an APK to test it on Android devices. As we share data with a web service in the proposed architecture, there are several security implications that need to be considered to protect user privacy. While the main focus of this paper is not the security aspect, we followed the best practices related to each software framework utilized. Transmitting sensitive user data over the Internet may expose it to eavesdropping by malicious actors, and there is also a risk of data tampering during transmission. Furthermore, the web service may become a target of DoS (Denial of Service) attacks and insecure communication channels may lead to man-in-the-middle attacks. As such, we followed a holistic approach, which involves secure coding practices, ongoing monitoring, and regular security assessments. For instance, the Passport library was used for the authentication process. Bcrypt was used for encrypting and decrypting user passwords when creating new accounts or during authentication when logging in. We

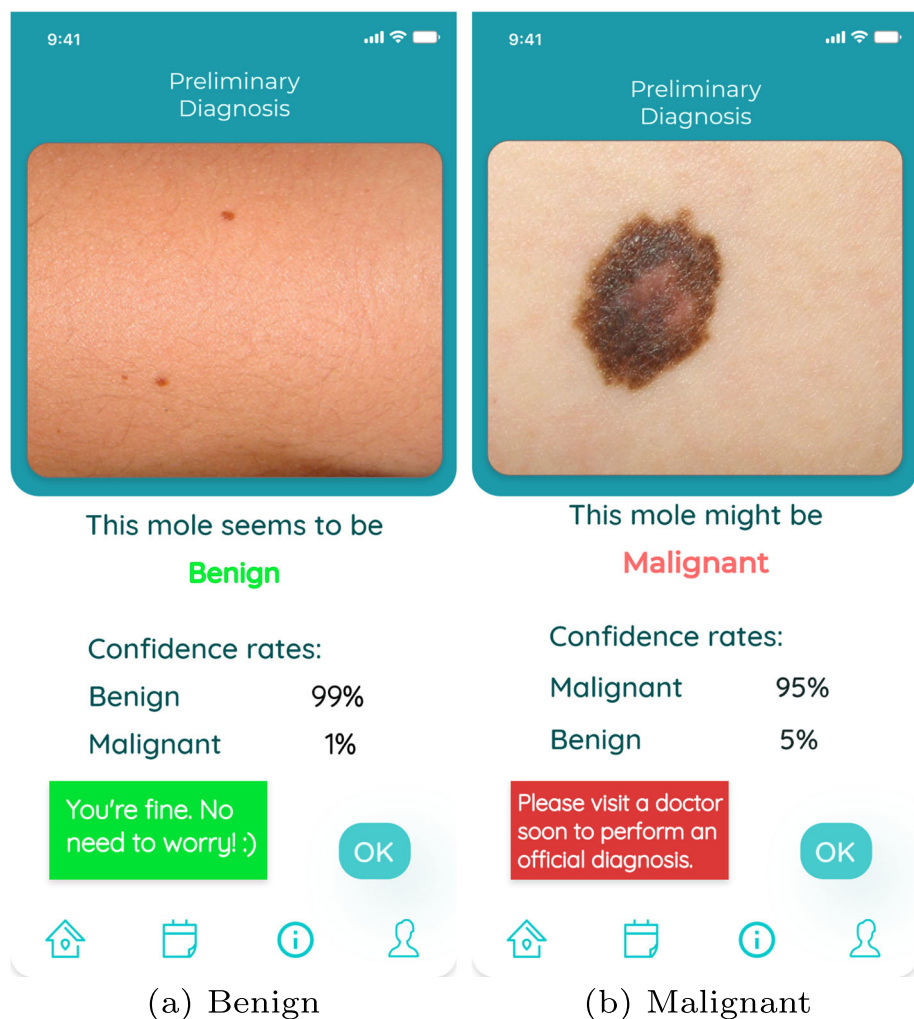


Fig. 9 Sample Screens for Classification Results

ensured that Node.js and Passport are configured securely, following the best practices. Also, we validated and sanitized all user input to prevent common security vulnerabilities such as SQL injection. We also keep Node.js, Passport, and other dependencies up to date with the latest security patches. As security is a continuous process, we must be proactive in addressing emerging security threats.

In the future, we aim to create a much larger dataset and include new models for benchmarking. We also want to focus on interpretable machine learning (a.k.a., explainable artificial intelligence) aspects. Another research direction can be related to digital twins, which have been applied successfully in many different domains [20–22]. A digital twin can continuously update its representation based on changes in the user's skin over time. This application can compare images of the user's skin with the digital twin to monitor any evolving moles. Different object detection algorithms can be investigated from a recent review article [23] and novel models can be built.

Funding Open Access funding provided by the Qatar National Library.

Data Availability The paper has no associated data to declare.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Yan W (2019) Technologies for primary health care help meet global goals. *IEEE Pulse* 10(3):15–18. <https://doi.org/10.1109/MPULS.2019.2911822>
2. Foster KR, Callans DJ (2017) Smartphone apps meet evidence-based medicine: the future of medicine may (or may not) be in your smartphone. *IEEE Pulse* 8(6):34–39. <https://doi.org/10.1109/MPUL.2017.2750783>
3. Mertz L (2020) Predictive models on the rise, but do they work for health care? *IEEE Pulse* 11(6):10–13. <https://doi.org/10.1109/MPULS.2020.3036151>
4. Gupta A, Thakur S, Rana A (2020) Study of melanoma detection and classification techniques. 2020 8th International conference on reliability, infocom technologies and optimization (trends and future directions) (ICRITO), pp 1345–1350. <https://doi.org/10.1109/ICRITO48877.2020.9197820>
5. Benbrahim H, Hachimi H, Amine A (2020) Deep convolutional neural network with tensorflow and keras to classify skin cancer images. *Scalable Comput Pract Exp* 21(3):379–390. <https://doi.org/10.12694/scpe.v21i3.1725>
6. Namozov A, Cho YI (2018) Convolutional neural network algorithm with parameterized activation function for melanoma classification. 2018 International Conference on Information and Communication Technology Convergence (ICTC), pp 417–419. <https://doi.org/10.1109/ICTC.2018.8539451>
7. Guha SR, Rafizul Haque SM (2019) Convolutional neural network based skin lesion analysis for classifying melanoma. 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), pp 1–5. <https://doi.org/10.1109/STI47673.2019.9067979>
8. Li H et al (2019) Dense deconvolutional network for skin lesion segmentation. *J Biomed Health Inform* 23(2):527–537. <https://doi.org/10.1109/JBHI.2018.2859898>
9. Yuan Y, Lo Y-C (2019) Improving dermoscopic image segmentation with enhanced convolutional-deconvolutional networks. *IEEE J Biomed Health Inform* 23(2):519–526. <https://doi.org/10.1109/JBHI.2017.2787487>
10. Song L, Lin J, Wang ZJ, Wang H (2020) An end-to-end multi-task deep learning framework for skin lesion analysis. *IEEE J Biomed Health Inform* 24(10):2912–2921. <https://doi.org/10.1109/JBHI.2020.2973614>
11. Budhiman A, Suyanto S, Arifanto A (2019) Melanoma cancer classification using resnet with data augmentation. 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp 17–20. <https://doi.org/10.1109/ISRITI48646.2019.9034624>
12. González-Díaz I (2019) DermaKNet: incorporating the knowledge of dermatologists to convolutional neural networks for skin lesion diagnosis. *IEEE J Biomed Health Inform* 23(2):547–559. <https://doi.org/10.1109/JBHI.2018.2806962>
13. Akay M et al (2021) Deep learning classification of systemic sclerosis skin using the mobilenetV2 model. *IEEE Open J Eng Med Bio* 2:104–110. <https://doi.org/10.1109/OJEMB.2021.3066097>
14. Ashraf R et al (2020) Region-of-interest based transfer learning assisted framework for skin cancer detection. *IEEE Access* 8:147858–147871. <https://doi.org/10.1109/ACCESS.2020.3014701>

15. Castro PBC, Krohling B, Pacheco AGC, Krohling RA (2020) An app to detect melanoma using deep learning: an approach to handle imbalanced data based on evolutionary algorithms. 2020 International Joint Conference on Neural Networks (IJCNN), pp 1–6. <https://doi.org/10.1109/IJCNN48605.2020.9207552>
16. Akbulut A, Perros HG (2019) Performance analysis of microservice design patterns. *IEEE Internet Comput* 23(6):19–27
17. Scarlat A (2018) Melanoma: augmented dermoscopic pigmented skin lesions from HAM10k. Version 1. [Dataset]. Available: <https://www.kaggle.com/drscarlat/melanoma/version/1>
18. SensioAI (2020) Melanoma224: dataset utilizado en los vídeos de sensioai. Version 1. [Dataset]. Available: <https://www.kaggle.com/sensioai/melanoma224/version/1>
19. Bhatt H, Shah V, Shah K, Shah R, Shah M (2022) State-of-the-art machine learning techniques for melanoma skin cancer detection and classification: a comprehensive review. *Intell Med Elsevier*
20. Shoukat MU, Yan L, Du C, Mair Raza MU, Adeel M, Khan T (2022) Application of digital twin in smart battery electric vehicle: industry 4.0. In: 2022 International conference on IT and industrial technologies (ICIT), pp 1–7. IEEE
21. Shoukat MU, Yan L, Liu W, Hussain F, Nawaz SA, Niaz A (2022) Digital twin-driven virtual control technology of home-use robot: human-cyber-physical system. In: 2022 17th International conference on emerging technologies (ICET), pp 240–246. IEEE
22. van Dinter R, Tekinerdogan B, Catal C (2022) Predictive maintenance using digital twins: a systematic literature review. *Inf Softw Technol* 107008. Elsevier
23. Nawaz SA, Li J, Aslam Bhatti U, Shoukat MU, Ahmad RM (2022) AI-based object detection latest trends in remote sensing, multimedia and agriculture applications. *Front Plant Sci* 13:1041514. Frontiers

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.