A Deep Learning Model for the Segmentation and Classification of Melanoma Skin Lesions

Ombaso Tony Mogoa
116814 ICS 4A

Supervisor Name
Orero, Joseph Onderi (Dr.)

Submitted in Partial Fulfilment of the Requirements of the Bachelor of Science in Informatics and Computer Science at Strathmore University

School of Computing and Engineering Sciences
Strathmore University
Nairobi, Kenya

December 2022

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other university. To the best of my knowledge and belief, the project documentation contains no material previously published or written by another person except where due reference is made in the project documentation itself.

Student Name: Ombaso, Tony Mogoa
Admission Number: 116814
Student Signature: ________________________ Date: ________________________

The project documentation of Ombaso, Tony Mogoa has been reviewed and approved by Dr Joseph Onderi Orero.
Supervisor Signature: ________________________ Date: ________________________

Acknowledgement

I would like to express my gratitude to my supervisor Dr Joseph Orero for guiding me throughout the whole journey of writing this project documentation.

Abstract

Melanoma is the most fatal of skin cancers as it accounts for most skin cancer deaths. However, if melanoma is detected early, it is highly curable. Its diagnosis involves non-invasive examination of skin lesions and their subjection to biopsy. The non-invasive examination can be challenging even for dermatologists hence requires a lot of experience. Machine learning models, trained on dermoscopic and clinical images, have been applied in classification of skin lesions to address the shortage of dermatologists and increase chances of early diagnosis. Though dermoscopy increases accuracy in diagnosis, it is more expensive and needs more technical expertise while clinical images, which are taken using user-grade cameras, are cheaper and more accessible hence have a greater potential of making melanoma diagnosis more accessible and of supporting self-examination to increase chances of early diagnosis. Existing models for classifying melanoma lesions from clinical images have questionable generalizability having been trained on very small datasets. This research used experimental methodology to develop a deep learning model for lesion segmentation and classification for increased generalizability through a comparative study of deep learning architectures involving training and testing across five non-dermoscopic datasets.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

API	—	Application Programming Interface
AUC	—	Area Under Curve
CNN	—	Convolutional Neural Network
ERD	—	Entity Relationship Diagram
ETL	—	Extract, Transform, Load
GPU	—	Graphical Processing Unit
HTTPS	—	Hypertext Transfer Protocol Secure
ML	—	Machine Learning
NLP	—	Natural Language Processing
OOAD	—	Object Oriented Analysis and Design
RESTful	—	Representational State Transfer - ful
UML	—	Unified Modelling Language
URL	—	Uniform Resource Locator
UV	—	Ultra Violet Radiation
VGG	—	Visual Geometry Group

Chapter 1: Introduction

1.1 Background Information

Melanoma is a type of skin cancer which develops due to the mutation of melanocytes which are specialised cells that produce melanin (**?**). Melanoma is the most fatal of skin cancers since while it accounts for 3% of skin cancers, it is responsible for 65% of deaths associated with skin cancer (**?**). However, according to **?**, the survival rate for melanoma cancer patients is high if diagnosed early since it can be cured through minor surgery alone, whereas if the cancer has already spread to other organs the survival rate drops significantly.

The diagnosis of melanoma involves the examination of pigmented lesions on the skin to determine whether they are malignant or benign followed by an excisional biopsy if there is doubt. Excisional biopsy refers to the removal of suspicious lesion tissues (**?**). The extracted tissues are then examined using a microscope. The two most common methods of non-invasive examination are dermoscopy and examination with the naked eye (**?**). Dermoscopy, more accurately called epiluminescence microscopy, is a non-invasive technology that by use of oil immersion eliminates the reflectance of light from the skin surface enabling the inspection of subsurface structures of the skin hence enhancing the diagnosis of pigmented lesions. Dermoscopy uses a piece of special equipment called a dermatoscope (**?**). Examination with naked eyes, on the other hand, does not involve the use of any special equipment, hence examination of skin lesions through dermoscopy can be more accurate and sensitive if used by trained clinicians (**?**). Once potentially malignant lesions have been identified, they may be taken for biopsy in the case of ambiguous lesions, where a definitive judgement is made on whether or not the patient has developed malignant melanoma since biopsy is still the gold standard in the diagnosis of melanoma (**?**).

The examination of skin lesions for the diagnosis of melanoma is crucial if malignant melanoma is to be determined early enough. According to **?**, the accurate identification of whether a skin lesion is malignant or benign can be challenging, making the misdiagnosis of melanoma a significant public health problem in the US. For this reason, this task requires experienced dermatologists who happen to be few (**?**), for instance, as **?**

estimate, in Africa there is, a dismal, less than one dermatologist per million population in urban areas. Patients may be forced to wait for a long time to be able to see an experienced dermatologist since there is a lot of workload on the few dermatologists (?). While patients bid their time in wait for a dermatologist, melanoma could be advancing. To address this shortage, computerised solutions for the diagnosis of melanoma have been developed and have been under active research. A vast majority of these solutions comprise deep learning models trained on dermoscopic images and on non-dermoscopic(clinical or macroscopic) images (?).

While dermoscopy has enhanced the accurate identification of malignant melanoma skin lesions, it requires specialized equipment and hence is more expensive and requires a lot of technical expertise (??). On the other hand, according to ?, clinical images incur lower cost and are easily available since they can be taken using user-grade cameras, albeit with non-uniform illumination, and may contain fewer details and poorer contrast as compared to dermoscopic images. Therefore models trained on clinical images stand a greater chance of being more accessible to low-income populations.

This research developed the development of a deep learning model using non-dermoscopic images, which users can use for self-examination to increase the chances of identification of early melanoma malignancy while reducing the workload on dermatologists. The model could also be used in dermatology clinics for prioritization in the scheduling of appointments with dermatologists (?). Additionally, the model can be used as a decision support tool for dermatologists to improve the accuracy of diagnosis hence potentially reducing the number of melanoma misdiagnoses and the number of unnecessary biopsies (?). However, predictions made by the model will neither be a medical diagnosis nor a substitute for it.

A vast majority of technological approaches to the diagnosis of melanoma from non-dermoscopic images have focused on machine learning techniques. ? use k-means clustering for lesion segmentation, SkinVision, a mobile application, uses Scalable Vector Machines (SVM) for classification of lesions, ? uses a six-layer CNN for classification, ? used a pre-trained ResNet-152 CNN to classify 12 skin cancer types and ? used a pre-trained GoogleNet Inception v3 CNN for classification. ? used an ensemble model which employs a genetic algorithm to select a subset of CNN architectures from a population, while **?** used VisionTransformer to classify skin lesions.

1.2 Problem Statement

The need for early detection of melanoma cannot be overemphasized. Though dermoscopy has enhanced the identification of malignant melanoma skin lesions, its use of a dermatoscope makes it expensive and requires technical expertise (**??**). Since patients are not expected to buy dermatoscopes and be trained on dermoscopy, a cheaper and more accessible alternative means is required. To address this need for accessibility, this research proposes the development of a deep learning model for the classification and segmentation of melanoma skin lesions from non-dermoscopic images, since non-dermoscopic images can be taken using user-grade cameras which are inexpensive (**?**).

Many researches have attempted to devise technological means for the identification of melanoma skin lesions with the most promising solution being the use of machine learning techniques. The generalizability of the models developed by existing researches for the classification of melanoma from non-dermoscopic images is questionable as **?** assert since only small and imbalanced datasets are available. For instance, the model developed by **?** when tested on a dataset unseen before, the Edinburgh dataset, shows a significantly lower performance. **?** and **?** use the MED-NODE dataset which has only 170 images. Additionally, **?** argues that available smartphone applications for the identification of malignant melanoma have been shown to be inaccurate by multiple studies.

This research, therefore, was aimed at doing a comparative study of various machine learning technologies across five non-dermoscopic datasets as listed in Table **??**, to enhance the generalizability of the developed model. This research compared the classification performance of VisionTransformer, ResNet34, Resnet50, Resnet152, InceptionV3 and EfficientNetV2 across the five datasets and proposed the most suitable model that enhanced generalizability in the segmentation and classification of melanoma skin lesions.

1.3 Objectives

1.3.1 General Objective

To develop a deep learning model for the segmentation and classification of skin lesions as either benign or malignant.

1.3.2 Specific Objectives

(i) To study and analyse current clinical practice in the diagnosis of melanoma and the associated challenges

(ii) To study and analyse the current technological solutions used in the diagnosis of melanoma

(iii) To study and analyse the challenges facing the existing technological solutions used in the diagnosis of melanoma

(iv) To design and build a deep learning model for segmentation and classification of skin lesions as either benign or malignant

(v) To test and validate the developed model

1.4 Research Questions

(i) What is the current clinical practice in the diagnosis of melanoma and what are the associated challenges?

(ii) What are the current technological solutions used in the diagnosis of melanoma?

(iii) What are the challenges facing the existing technological solutions used in the diagnosis of melanoma?

(iv) How will a deep learning model for segmentation and classification of skin lesions as either benign or malignant be designed and built?

(v) How will the developed model be tested and validated?

1.5 Justification

Since dermoscopy is expensive and the early diagnosis of melanoma crucial, the population needs to be equipped with a means of identifying melanoma cheaply and accurately given that there is a shortage of dermatologists (**???**). According to **?** current technological solutions for the diagnosis of melanoma from clinical skin lesion images have been shown to be often inaccurate by multiple studies. According to **?** the availability of large datasets in the medical domain is rare hence there is the danger of poor generalizability due to overfitting. The studies produced by **?** and **?**, utilize the MED-NODE dataset (**?**), a small dataset which has only 170 images while the model proposed by ? shows a significantly lower performance when evaluated on the Edinburgh dataset which was not used for training the model. Hence the generalizability of their proposed models is questionable. Therefore, a more generalizable study was needed hence this research was aimed at developing a deep learning model for the segmentation and classification of melanoma lesions through a comparative study involving various deep learning architectures across five non-dermoscopic datasets.

1.6 Scope and Delimitations

This study was limited to the classification of melanoma from clinical (non-dermoscopic) images as opposed to dermoscopic images. This is because dermoscopic images require the use of a dermatoscope which is expensive and requires a lot of expertise while clinical images only require a user-grade camera hence cost less and are more accessible in areas with a shortage of specialised dermatological care ?. Additionally, the development of a model for the classification of melanoma skin lesions was intended to give users a medical diagnosis, the model was only to inform the user of the level of risk of a lesion being melanoma. Neither was it intended to be a substitute for medical diagnosis by a dermatologist.

1.7 Limitations

This study was limited firstly due to the absence of publicly available large enough non-dermoscopic(clinical) skin image datasets. Hence to prevent overfitting of the deep learning model, an ensemble dataset comprising datasets listed in Table ?? was used for training and testing. Data augmentation techniques were used to expand the ensemble dataset to prevent overfitting. Additionally, since the datasets to be used were acquired from white populations, the generalizability of this study was limited to white-skinned people.

Chapter 2: Literature Review

2.1 Introduction

This chapter discusses the literature review starting with a review of the current clinical practice in the diagnosis of melanoma. It goes on to discuss the challenges facing the diagnosis of melanoma. Furthermore, related works in the diagnosis of melanoma are discussed followed by a discussion of the gaps identified in existing works. Finally, the conceptual framework for the research is presented and discussed.

2.2 Current Clinical Practice in Diagnosis of Melanoma

Melanoma is a type of skin cancer which is caused by the mutation of melanocytes which are pigment-producing cells in the skin (**?**). If melanoma is discovered early there are high chances of survival since patients can be treated through minor surgery whereas if the cancer has already spread to other organs the survival rate is greatly reduced, therefore it is crucial that it is diagnosed early enough **?**.

The diagnosis of melanoma involves the examination of lesions (moles) on the skin of the patient. The dermatologist examines the moles to determine whether the lesion is benign, that is harmless, or whether it is malignant, meaning that it is cancerous (**?**). After the examination, if any lesion is suspected to be malignant, it is subjected to an excisional biopsy, where some tissue of the lesion is extracted for examination under a microscope which is referred to as histological examination. Histological examination is still the gold standard in the certain diagnosis of melanoma (**?**).

The examination of skin lesions before a biopsy is important since the number of unnecessary biopsies needs to be reduced. The two most common ways of pre-biopsy examination are examination with the naked eye and examination using a dermatoscope (**?**). The dermatoscope is a specialised instrument which uses oil immersion technology to the eliminate reflectance of light on the skin surface thereby making the subsurface structure of the skin available for microscopic examination (**?**). The use of a dermatoscope, that is dermoscopy, therefore enhances the diagnosis of melanoma.

Various clinical metrics for evaluating melanoma skin lesions have been developed such as the ABCDE rule and the 7-point checklist, to add objectivity to lesion examination.

The ABCDE technique offers an easy mnemonic for evaluating melanoma skin lesions, A for asymmetry, B for border irregularity, C for colour variegation, D for diameter generally greater than 6mm and E for evolution (lesions changing over time) (?). The 7-point checklist relies on the use of a dermatoscope, evaluating skin lesions against the following items: atypical pigment network, grey-blue areas, atypical vascular pattern, radial streaming, irregular diffuse pigmentation, irregular dots and globules, and regression pattern (?). According to ?, these clinical metrics have improved the accuracy, specificity and sensitivity of the diagnosis of melanoma.

2.2.1 Challenges Facing Diagnosis of Melanoma

Misdiagnosis remains to be a challenge in the diagnosis of melanoma since according to ? malignant melanoma can mimic benign lesions and benign lesions, in turn, can appear to be melanomas. For this reason, every dermatologist bears the anxiety of potential misdiagnosis, it is therefore not surprising that misdiagnosis of melanoma is one of the most common causes of medical litigation (?). Despite the development of dermoscopic clinical metrics, ? report that both the ABCDE rule and 7-point checklist have inadequate diagnostic accuracy for early melanomas with Area Under Curve (AUC) scores of 0.67 and 0.64 respectively with both showing sensitivities of less than 32%. ? in a more recent study, report even lower AUC scores (0.57 and 0.59 respectively) while indicating that there is no universally accepted clinical metric to date. Therefore, it follows that more accurate techniques need to be developed.

According to ?, developing proficiency in diagnosis through dermoscopy requires a significant amount of time and effort from dermatologists. Moreover, dermatoscopes do not come cheap (?). For this reason, dermoscopy is not ideal in enabling patients to carry out self-examination of skin lesions and much less so given that it requires a lot of skill and experience even for dermatologists. Given that researches on the effectiveness of dermoscopy seem to be contradicting each other on whether dermoscopy increases diagnostic accuracy (??), it can be concluded that examination with the naked eye is still a valid option, more so in the view of dermoscopy sceptics.

Early detection of melanoma is still a challenge in dermatological care (??). According to ?, enabling patients to recognize melanoma and seek dermatological care is crucial for the early diagnosis of melanoma. ? point out that conditions such as nodular melanoma are difficult to clinically diagnose even with dermoscopy, nevertheless noting that the recent advances in artificial intelligence incorporated in smartphone applications could potentially improve early diagnosis.

The shortage of dermatologists and specialized dermatology care most especially in developing countries is a concern. According to ?, most African countries have a ratio of less than one dermatologist per million population. This shortage means delayed diagnosis of melanoma since the few dermatologists are overburdened and subsequently potentially higher mortality rates.

2.3 Related Works

Related works in the technological diagnosis of melanoma have revolved around using dermoscopic and clinical (non-dermoscopic) images. The various technologies used in the diagnosis of melanoma from non-dermoscopic and dermoscopic images are discussed below.

2.3.1 Mobile and Web Applications

A select few of the numerous mobile and web applications developed for the diagnostic evaluation of melanoma skin lesions are discussed in this section. However, it is interesting to note that most of the mobile applications published are not backed by scientific research (?).

2.3.1.1 SkinVision

Perhaps the most successful mobile app for melanoma diagnosis, SkinVision has more than 900,000 users worldwide and has been reviewed by multiple scientific studies (??) both reporting that SkinVision has a sensitivity of 95% and specificity of 78%. SkinVision is designed to detect three types of skin cancer malignant melanoma (MM), squamous cell carcinoma (SCC), and basal cell carcinoma (BCC). According to ?, the current version of SkinVision uses a conditional adversarial network for lesion segmentation combined with

SVM for classifying skin lesions. SkinVision also provides skin cancer information, Ultra Violet (UV) radiation updates, skin type and risk profile assessment.

2.3.1.2 UMSkinCheck

Developed by the University of Michigan, UMSkinCheck is a mobile application developed to help patients track lesions located on their body (**?**). It also provides information about melanoma to enhance awareness. Additionally, UMSkinCheck provides a risk calculator that assesses the risk of developing melanoma based on a series of questions regarding risk factors such as UV radiation.

2.3.1.3 SpotMole

SpotMole is a mobile application for classifying moles from images taken by a user, using the descriptors of the ABCD rule that is asymmetry, border irregularity, colour and diameter **?**. The publishers of the application claim to use machine learning techniques to analyse images but have no scientific research to speak of. **?** and **?** have used SpotMole as a benchmark each reporting a diagnostic accuracy of about 0.67 for SpotMole.

2.3.1.4 Skinensemble.com

Skinensemble.com is a web application which is an implementation of the research done by **?**. The web application allows users to upload lesion images, which can be segmented and classified as either benign or malignant. The classification is performed using a genetic deep learning model which uses an ensemble of CNN architectures. The model is discussed below under classification and segmentation models.

2.3.1.5 eSkin

? propose a mobile application that is notable for its use of a dermatoscope. A dermatoscope is connected to the phone and is used to take dermoscopic images whose lesions are segmented and classified using SVM. Their developed system achieves an accuracy of 92% (**?**).

2.3.2 Classification and Segmentation Models

2.3.2.1 Simple Machine Learning Models

? while introducing the MED-NODE dataset composed of non-dermoscopic images, used k-means clustering for lesion segmentation. After segmentation three different classifiers were employed: the prototype-based Cluster-based Adaptive Metric (CLAM) classifier which uses colour descriptors to classify lesion images, the Unbiased Color Image Analysis Learning Vector Quantization (UCIA-LVQ) which classifies lesions based on colour-texture descriptors and lastly the naive Bayesian Classifier which classifies lesions based on binary feature vectors based on presence of 10 attributes devised by ?. The final prediction is based on a majority vote of the three classifiers. The proposed MED-NODE system achieves a balanced accuracy score of 0.81.

2.3.2.2 CNN

? used a six-layer CNN with 2 convolving layers with a pooling layer after each convolving layer, to classify lesions in non-dermoscopic images using the MED-NODE dataset achieving specificity, sensitivity and accuracy of 0.80, 0.81, 0.81 respectively. ? produced an influential study in this area of research, where they fine-tuned a pre-trained deep residual CNN model ResNet-152 (?) using the non-dermoscopic Asan, MED-NODE, and Atlas Site datasets, publishing magnificent results for all the 12 skin cancer types. For melanoma they achieve an AUC score, sensitivity and specificity of 0.96, 91.0 ± 4.3, 90.4 ± 4.5 respectively, evaluated on the Asan dataset and 0.88 ± 0.01, 85.5 ± 2.3, 80.7 ± 1.1 respectively, for the commercial Edinburgh dataset (?).

? use a pre-trained fully convolutional neural network avoiding segmentation procedure for the reason that segmentation is a non-trivial task whose result can cause a misdiagnosis if important features are left out. ? achieve an accuracy of 81.8% over the 10-class, commercial Edinburgh dataset. More recently, ?, performing lesion segmentation using the Chan-Vese segmentation algorithm, use an ensemble model built via a genetic algorithm to classify melanomas from both dermoscopic and non-dermoscopic images. The genetic algorithm uses evolutionary computing to select suitable CNN models from a population of DenseNet, InceptionV3, MobileNet, NASNet, VGG and Xception. The selected suitable models form the ensemble model for the classification of melanomas. The study achieves a Mattews Correlation Coefficient of 0.954 on the MED-NODE dataset which was the only non-dermoscopic dataset among 15 dermoscopic datasets.

?, in an influential study, achieves remarkable results using the pre-trained GoogleNet Inception v3 CNN trained on 129,450 clinical images (not available in a public dataset) reporting an AUC score of 0.94 for melanoma diagnosis. They can overcome problems of non-uniform zoom, angle and lighting inherent in clinical images by using Inception v3 which is pre-trained on approximately 1.28 million images. **?**, using a dermoscopic dataset, used a fully convolutional U-Net CNN for segmentation, used Caffe CNN, deep residual network and U-Net to extract features and used SVM to classify the extracted features.

2.3.2.3 SVM

? used a pre-trained CNN model, VGG16, to extract features from the non-dermoscopic images of the MED-NODE dataset and applied a neutrosophic logic support vector machine to classify the lesions to achieve an accuracy score of 0.91 and F1 measure of 0.93 evaluated on the MED-NODE dataset. Moreover, **?** used SVM to classify lesions based on features extracted from an ensemble of CNNs using a dermoscopic dataset. Both **?** and **?** use SVM on a private non-dermoscopic dataset in the classification step, the former using CAN for lesion segmentation and the latter using a seeded region-growing algorithm.

2.3.2.4 VisionTransformer

With the noteworthy success of transformer neural networks in NLP, their use in computer vision has been explored. **?**, in an influential study, show that pure Vision Transformers perform very well in image classification tasks with substantially lesser computational resources. **?** use VisionTransformer in the classification of non-dermoscopic skin lesions, trained and evaluated on the PADUFES-20 dataset achieving state-of-the-art results of the balanced accuracy metric.

2.3.2.5 Adversarial Networks

To address the shortage of data, especially for non-dermoscopic images, some studies have opted to use Generative Adversarial Networks (GANs) to augment the small datasets that are available. **?** propose a Self-attention Progressive Growing of GANs (SPGGANs) governed by the Two-Timescale Update Rule (TTUR) to overcome the challenge of unstable training which is associated with conventional GANs. The SPGGAN is used to generate skin lesion images using the dermoscopic HAM10000 skin lesion dataset, reporting a significant improvement in sensitivity over non-augmented(by 13.8%) and classically augmented (by 8.6%) counterparts. Despite these improvements, GANs are marred with the challenges of frequently failing to converge hence model instability and being expensive to tune (**?**). **?** use Conditional Adversarial Networks to perform lesion segmentation achieving a Jaccard index of 0.89 for pigmented lesions.

2.4 Gaps in Related Works

Choosing one type of lesion image over the other involves making trade-offs since on one hand dermoscopic images are taken at microscopic scales with uniform illumination thereby exposing more details of suspicious lesions as compared to clinical (macroscopic) images which are taken using normal cameras (**?**). On the other hand, clinical images have the advantage of being more accessible because they are less expensive as compared to dermoscopy which is not only expensive due to its use of special equipment but also requires high-level expertise (**??**). Therefore, models trained on dermoscopic images have limited accessibility since dermatoscopes are expensive, while non-dermoscopic images can be taken using standard cameras hence, models trained on them can be used by users for self-diagnosis thereby potentially increasing the chances of early diagnosis. For this reason, this research focuses on using non-dermoscopic images for melanoma diagnosis.

Various studies conducted on the effectiveness of smartphone applications for the diagnosis of melanoma have shown that they pose a risk of misdiagnosis, indicating that existing models are not accurate enough despite publishing remarkable results (**??**). For instance, SkinVision has been shown to have a specificity of 78% which as **?** acknowledges needs to be improved. According to **?**, the vast majority of mobile apps published have no scientific research backing them.

There also remains the problem of insufficient data owing to the smallness of public non-dermoscopic datasets. It is therefore possible that published models have overfitted the small datasets as **?** point out. For instance, the model built by **?**, performed significantly lesser tested on the commercialised Edinburgh dataset which did not form part of the training set. In as much as the use of GANs to generate non-dermoscopic lesion images has been shown to improve diagnostic accuracy, GANs are generally hard to train properly (**?**). Even more reason for their unsuitability, at least presently, is whether there would be medical accreditation for the use of GANs to generate lesions images given that the classification of melanoma is a non-trivial task with a huge impact on the diagnosed patient.

To improve the external validity of this study given the limited availability of large enough datasets (**?**), this research was aimed at using an ensemble dataset comprising the MED-NODE dataset, PAD-UFES-20 dataset, DermIS dataset, DermQuest dataset and the Seven-Point Checklist Dermatology Dataset(7-point dataset). Table **??** shows a list of the datasets used, the publishers and the number of images they had.

Dataset	**Publisher**	**No. of images**
MED-NODE	**?**	170
PAD-UFES-20	**?**	296
7-point	**?**	827
DermIS	**?**	69
DermQuest	Galderma S.A.	137
	Total	**1499**

Table 2.1: Table showing list of datasets used

2.5 Conceptual Framework

Figure ?? shows the conceptual framework of the developed solution. All five datasets, as listed in Table ??, were pre-processed which entailed resizing and centre-cropping to achieve uniformity. Data augmentation was then applied to expand the dataset by applying image manipulation techniques such as cropping, zooming, rotation and stretching. The lesions in the resulting set of images were then segmented using U-net CNN and used for training. The U-Net CNN was trained using the DermIS and DermNet datasets which provided binary segmentation masks for each lesion. Each of the machine learning models namely Visual Transformer, ResNet34, ResNet50, ResNet152, InceptionV3 and Efficient-B7 were trained using lesion images resulting from the segmentation step and then tested through using a held-out validation set. The performance of the machine learning models was then compared and the most suitable model was picked and used to develop a mobile application for classifying melanoma skin lesions.

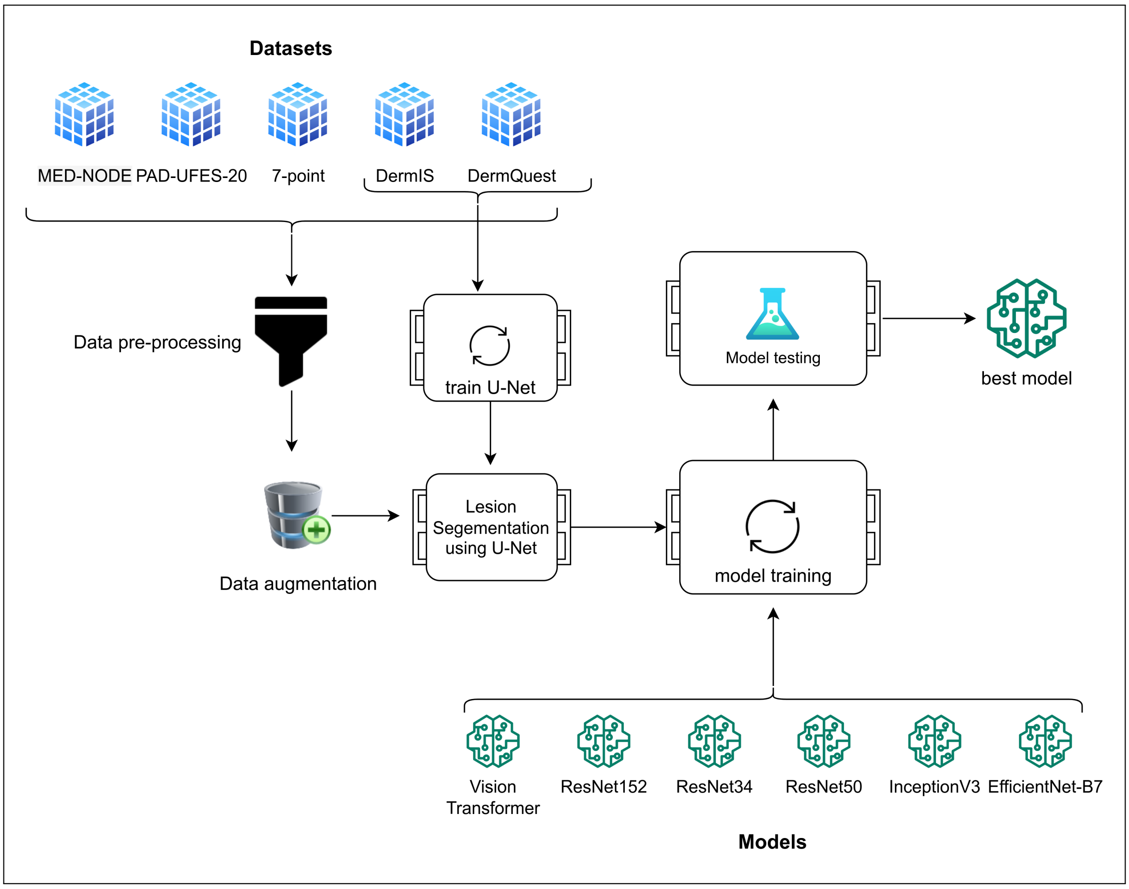


Figure 2.1: Conceptual framework

Chapter 3: Methodology

3.1 Introduction

This chapter discusses the design paradigm and methodology used in the development of the developed solution. Various analysis and design diagrams used are outlined, deliverables for the developed solution are discussed and finally, the tools and techniques used in development are presented.

The design paradigm used is the Object-Oriented Analysis and Design (OOAD) paradigm. OOAD is a design approach that simplifies real-world systems by using the concept of objects and their interaction to model systems in software development (**?**). OOAD was more suited to the development of the solution since the design of the system is greatly simplified by conceptualization using objects, and OOAD is flexible as it allowed the developed solution to be iterated on to produce an optimal version of the solution.

3.2 Methodology

The methodology applied was the incremental development methodology. According to **?**, the incremental development methodology involves the development of an initial implementation of a solution followed by the evolution of that implementation, also called an increment, through a series of iterations till an optimal satisfaction of the requirements is obtained. Figure **??** below shows a diagrammatic depiction of the incremental development methodology.

The incremental development methodology was suited to the development of the developed solution since the nature of the solution demands iteration. Deep learning involves training models over numerous epochs and tuning hyperparameters hence requiring a lot of iterations. The incremental development methodology involves three major activities which are discussed below namely, specification, development and validation. These activities were repeated with each iteration.

3.2.1 Requirements Specification

The requirements specification phase involves the collection of the requirements for the system and their analysis. In this phase, requirements for the developed solution were defined through document analysis of existing technological solutions. With every subsequent iteration, this phase was repeated.

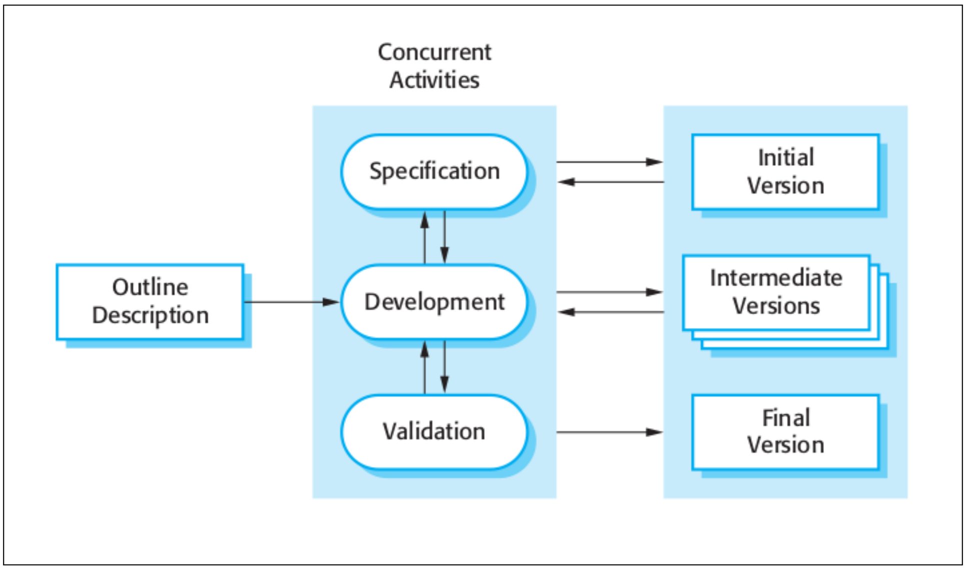


Figure 3.1: Depiction of the incremental methodology from (?)

3.2.2 Development

The development phase involves the designing and implementation of a system guided by the requirements specification. For each iteration, during this phase, the developed solution was designed and built as per the requirements specified in the previous phase.

3.2.3 Validation

The validation phase involves ascertaining that the implementation of the development phase does indeed conform to the requirements specification guiding its development. The ascertainment was done through testing. Each increment was validated during this phase.

3.3 Analysis Diagrams

3.3.1 Use Case Diagram

A use case diagram is a diagram that shows the high-level functions of a system and how various actors interact with those functions. The use case diagram was used to analyse the actors and functions of the developed system.

3.3.2 Entity Relationship Diagram

An Entity Relationship Diagram (ERD) is a diagram that depicts the high-level entities in a system and the relationship among them. The ERD diagram was used to analyse the entities involved in the developed solution, their relationships and cardinality.

3.3.3 Sequence Diagram

A sequence diagram is an object-oriented system modelling diagram that shows the sequential interaction of objects within a system. The sequence diagram was used to analyse how the objects in the developed solution were to interact to successfully segment and classify skin lesion images.

3.3.4 System Sequence Diagram

A system sequence diagram is an object-oriented system modelling diagram that shows the sequential interaction of external actors with a system. The sequence diagram was used to analyse how the external actors were to interact with the developed solution to successfully segment and classify skin lesion images.

3.3.5 Activity Diagram

An activity diagram is a Unified Modelling Language (UML) diagram that depicts the flow of activities of events of a system for a given use case or multiple use cases. The activity diagram was used to analyse the flow of activities and events during the segmentation and classification of a skin lesion image.

3.3.6 Class Diagram

A class diagram is a UML diagram that models a system by depicting the system's classes, class attributes, class operations and the relationship between the objects defined by the classes. The class diagram was used to analyse how the entities of the developed system would be mapped into classes and the relationship between the classes.

3.4 Design Diagrams

3.4.1 Database Schema

A database schema diagram depicts the structure of tables which correspond to entities in a system and the relationship between the tables. The database schema diagram was used to design a suitable database design for the developed system.

3.4.2 Wireframes

Wireframes are line drawings which depict how the user interface of a system will be structured. Wireframes were used to design the user interface and simulate the user experience of the developed system.

3.4.3 System Architecture Diagram

A system architecture diagram is a diagram that depicts the interaction of the top-level independent functional units of a system. The system architecture was used to design the functional units of the developed solution and to profile their interaction.

3.5 Deliverables

3.5.1 Research Proposal

A research proposal is a formal document that outlines what research is about by stating the title, discussing background information of the area of research, stating objectives, justification, scope and limitations, reviewing the literature on the area of research and discussing the methodology for the research project. The research proposal justified the research and guided the implementation of the developed solution.

3.5.2 Project Documentation

The project documentation is a formal document that outlines what the project is about by stating the title, discussing background information of the area of research, stating objectives, justification, scope and limitations, reviewing the literature on the area of research, discussing the methodology for the research project, implementation and testing of the developed system and ends with conclusions, recommendations and points out future works.

3.5.3 Deep Learning Model

A deep learning model refers to a deep neural network (a network with many layers of perceptrons) whose weights have been trained to perform a classification or regression task. The deep learning model was the core of the developed solution as it was responsible for the segmentation and classification of skin lesion images. The model was deployed on a server and exposed as an Application Programming Interface (API).

3.5.4 Mobile App User Interface

A mobile app was developed to provide a user interface through which users can use the deep learning model to classify skin lesion images. The app worked by accessing the camera of a mobile device to take a photo of a skin lesion image and then uploading the photo to the API hosting the deep learning model. Once the deep learning model determined the verdict, the mobile app notified the user of the results of the classification.

3.6 Tools and Techniques

3.6.1 Python

Python was used as the programming language in the implementation of the developed solution. This was because the majority of the machine learning and data science community used Python as the programming language of choice (**?**). Hence many indispensable tools for machine learning and deep learning supported python more than other languages.

3.6.2 PyTorch

PyTorch is a machine-learning framework maintained by Facebook. PyTorch was suited to this project compared to other machine learning frameworks since according to **?** it was gaining more popularity in research being published meaning it was deemed as better suited for research. PyTorch also provided more ease of development compared to other machine learning frameworks since it is more ”pythonic” and uses a more object-oriented style (**?**).

3.6.3 Google Colaboratory

Google Colaboratory, also known as Google Colab, was used to execute the Python code that implemented the developed solution. This is because Google Colab is a free Software as a Service(SaaS) application by Google that offers free Graphics Processing Units (GPU) and hence was used to efficiently train the machine learning models of the developed solution.

3.6.4 Expo

Expo was used to develop the mobile application. Expo is a framework built on top of React Native which exposes an API for building cross-platform mobile applications using purely JavaScript while abstracting the complexity of having to write native code on Xcode or Android Studio. React Native, released by Facebook is a framework that allows building cross-platform mobile applications using React syntax. Expo was used since it allowed the development of a cross-platform mobile application.

3.6.5 Data Augmentation

According to **?**, data augmentation is a technique of expanding the size of a dataset by applying several random transformations to original images. Since non-dermoscopic datasets were small there was a need to augment them to reduce overfitting. **?**, in a comparative study showed that all the models used in the study substantially performed better with data augmentation.

3.6.6 Transfer Learning

According to **?**, transfer learning is a technique of reusing already trained models from a source task which has a lot of data for a different but related task. **?**, **?** and **?** all use transfer learning in the form of pre-trained deep learning models showing a performance improvement. Therefore transfer learning was applied in this study to increase model performance.

Chapter 4: System Analysis and Design

4.1 Introduction

This chapter discusses how the system exposing the developed model was analysed and designed. Functional and non-functional system requirements are discussed and OOAD paradigm system analysis and design diagrams are presented and elaborated upon.

4.2 System Requirements

The system requirements for the developed system are as follows.

4.2.1 Functional Requirements

Functional requirements refer to the set of operations that a system must be able to provide (?). The functional requirements of the developed system include:

i. Authentication Module — This module provided registration and login functionality to users. The authentication module was needed so that users can keep records of their transactions with the system.

ii. Image Capture and Upload Module — This module was implemented in a mobile application to enable users to take a picture of a skin lesion and upload it to a server hosting the developed model for inference.

iii. Lesion Inference Module — This module consisted of the developed model which would take a lesion image as input and perform inference by segmentation followed by classification of whether the lesion is malignant melanoma or a benign mole. This module also included a user interface in the mobile app that will relay the results of the inference to the user.

4.2.2 Non-Functional Requirements

Non-functional requirements refer to the set of characteristics, attributes or constraints of a system which are desirable but not necessary for the functioning of a system (?). The non-functional requirements for the developed system include:

i. System Security — Given that the system dealt with medical data which tends to be sensitive, maximum privacy needed to be ensured. Passwords were hashed during the onboarding of users to guard against password breaches. All communication between the server and the client app was restricted to being over HTTPS only, hence ensuring that all data was encrypted in transit.

ii. System Performance — The system was built to minimise response time to requests as much as possible, thus ensuring a delightful user experience.

4.3 System Analysis Diagrams

4.3.1 Use Case Diagram

Figure **??** shows the use-case diagram for the developed system. The actors are the user and a Machine Learning (ML) model. The user can register with the system, login into the system, take a photo of a lesion and upload an image of a lesion upon which the lesion image is normalized in preparation for lesion inference. The ML Model can perform lesion classification on an uploaded image. Lesion classification includes lesion segmentation.

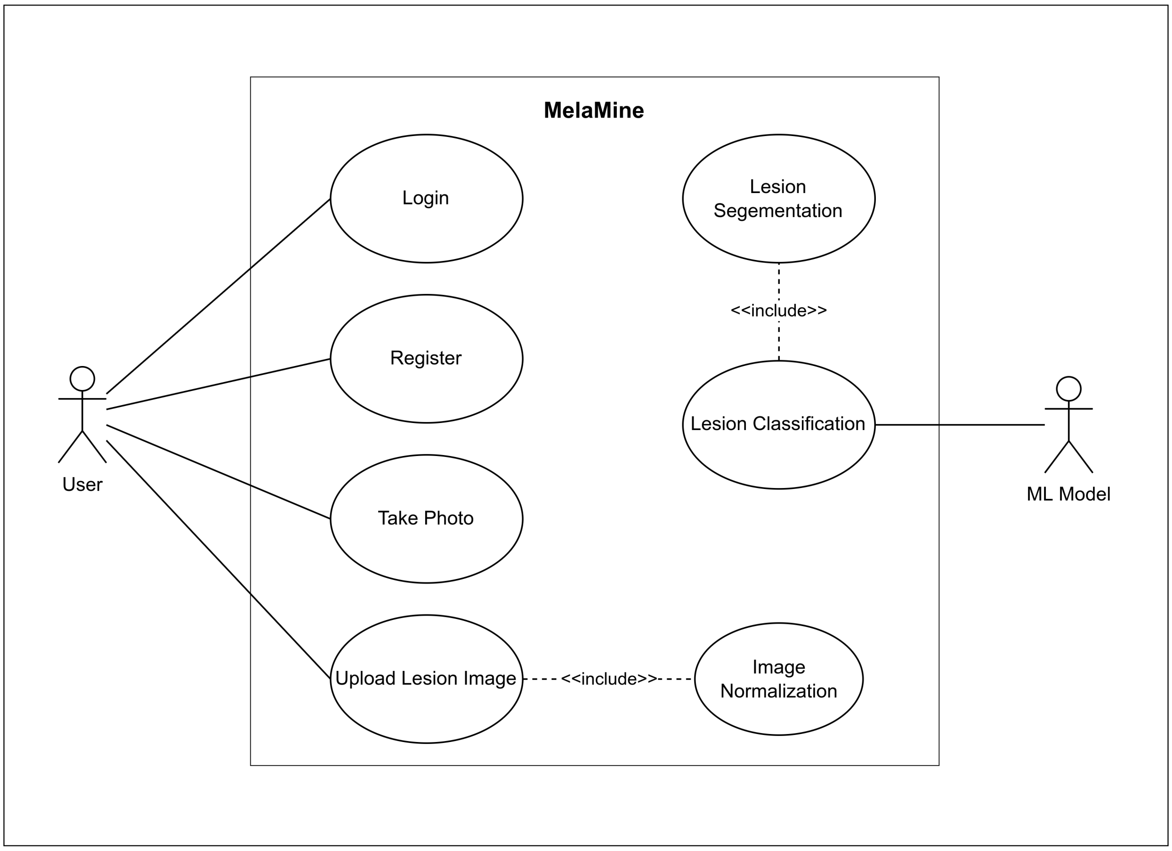


Figure 4.1: Use case diagram

4.3.2 Entity Relationship Diagram

Figure **??** shows the ERD for the developed system. The developed system consists of three top-level entities namely, the user, a lesion and an ML model. The user has the unique identifier 'userId', an email and a password. A lesion has the following attributes: the unique identifier 'lesionId', an image file of the lesion, a malignancy truth value and a timestamp. A user bears zero or more lesions and the ML model classifies zero or more lesions as malignant or benign.

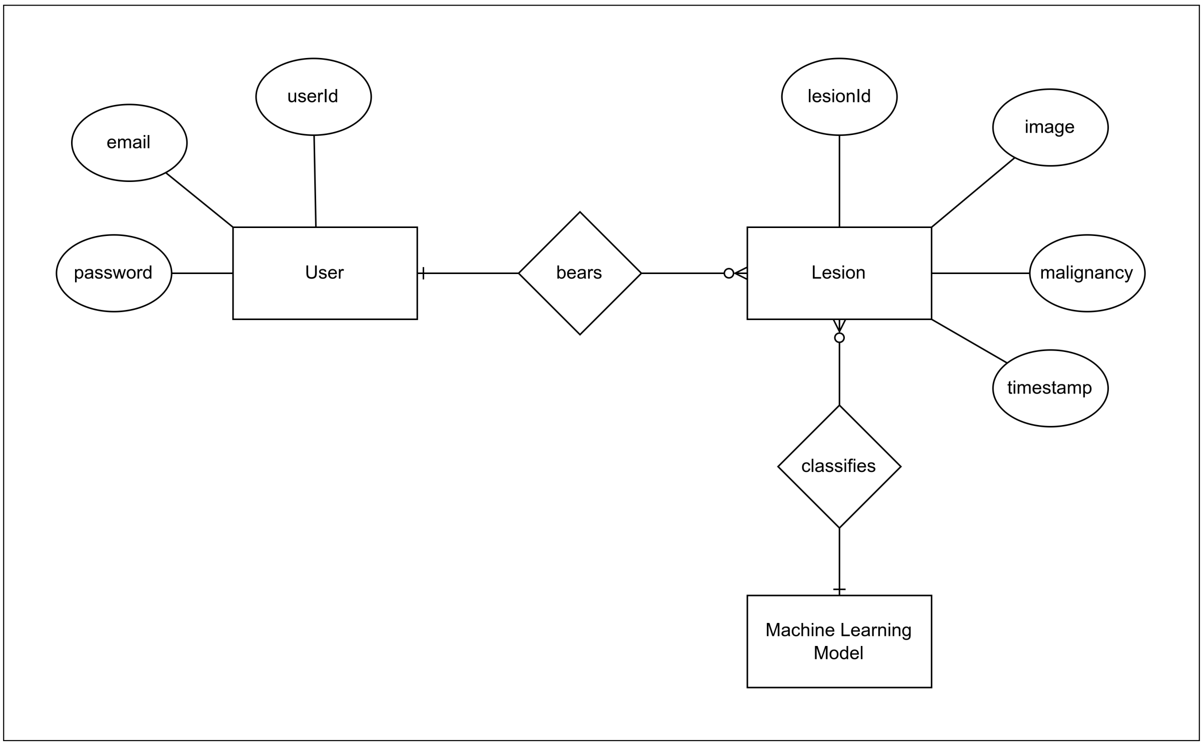


Figure 4.2: Entity Relationship Diagram

4.3.3 Sequence Diagram

Figure **??** shows the sequence diagram for the developed system. It depicts how the instances of User, Lesion and Model classes interact during lesion inference. The user object invokes the lesion object's setLesionImage method hence setting the image of the lesion to be classified. Subsequently, the user object invokes the isMalignant method of the lesion object and awaits for an inference result.

The lesion object subsequently invokes normalizeImage method to normalize the lesion image and calls the model object's getInference method. Consequently, the model object invokes its segment method and uses its classify method to classify the lesion and returns the inference result to the lesion object which in turn returns the inference result to the user object.

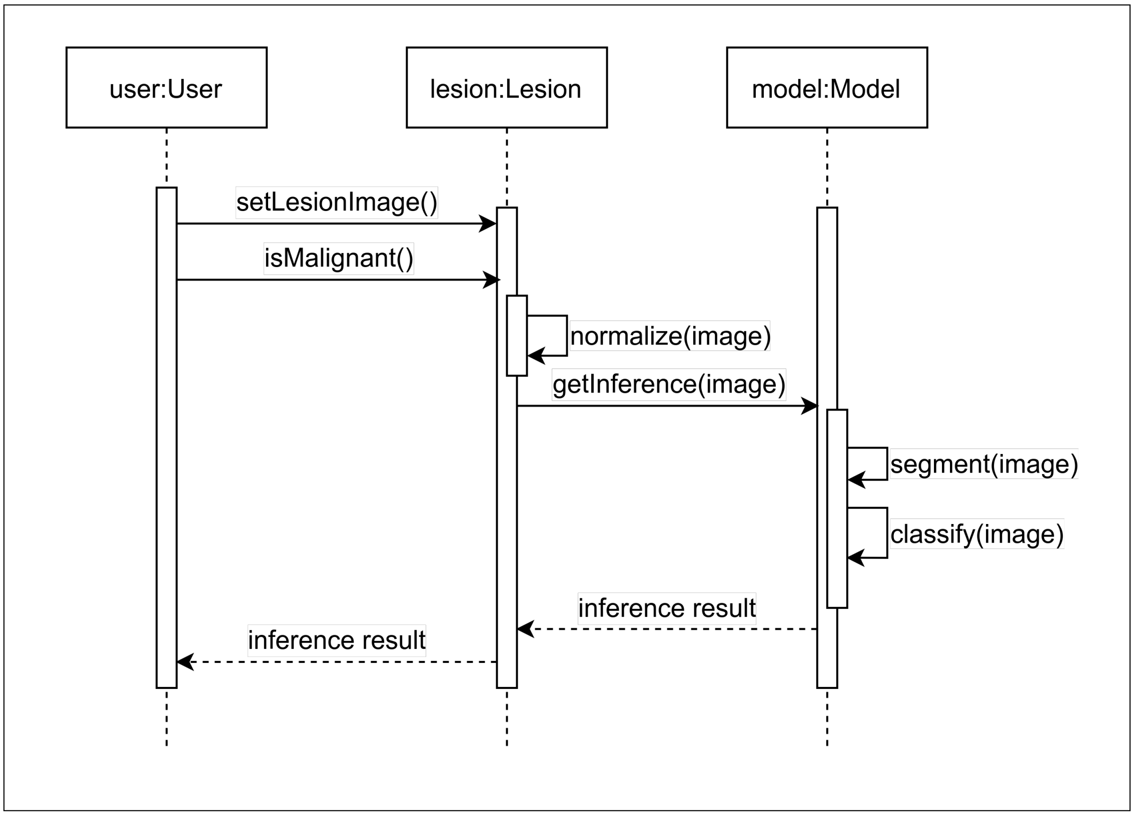


Figure 4.3: Sequence diagram

4.3.4 System Sequence Diagram

Figure **??** shows the system sequence diagram for the developed system. The sequence diagram depicts how a user can classify a lesion. The user wishing to log in is authenticated using an email and password combination by making a login request to the web server through the mobile app. If the credentials are correct the login is successful and the user takes a photo of a lesion or selects an existing image. The user, through the mobile app, then uploads the lesion image to the server. The server normalizes the image through resizing, cropping and transformation. The normalized image is then segmented and classified by the ML Model which then passes the result of inference back to the server which in turn relays the inference result back to the user through the mobile app.

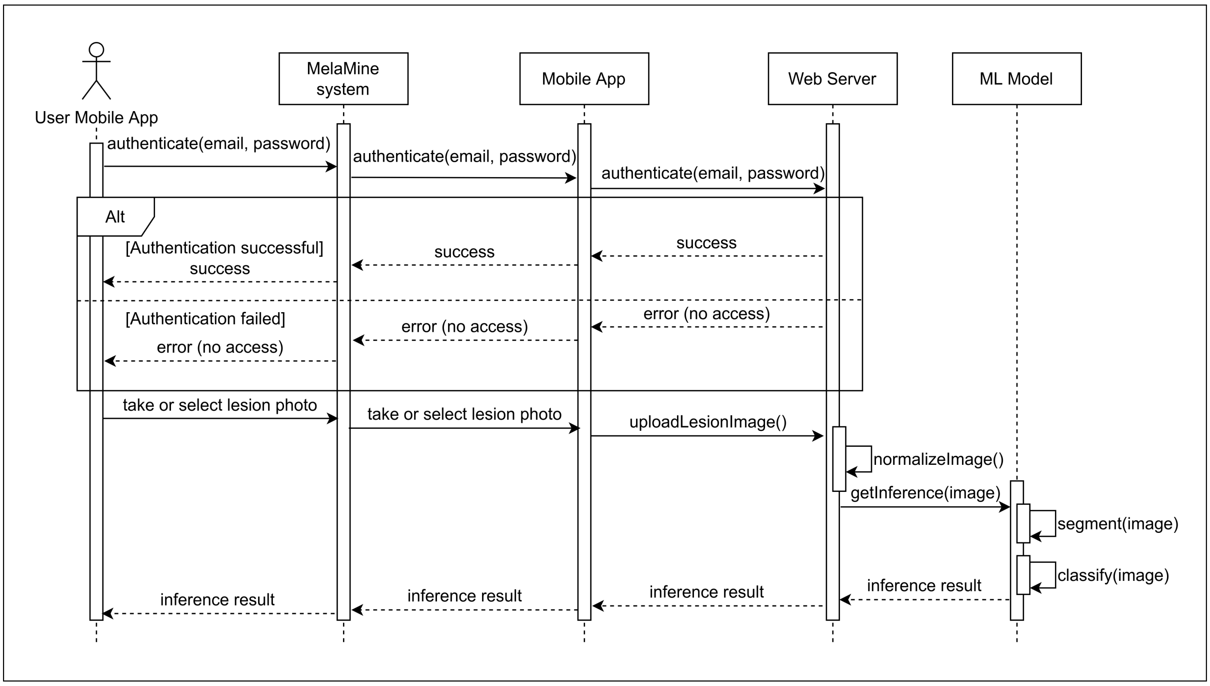


Figure 4.4: System sequence diagram

4.3.5 Activity Diagram

Figure **??** shows the activity diagram for the developed system. The activity diagram describes the activity by which a user can obtain inference for a skin lesion. There are four actors namely, the user, the mobile app, the web server and the ML model.

At the start of the activity, the user opens the mobile application and attempts a login action which succeeds if the user enters a valid username and password combination but fails otherwise. If the login attempt is successful, the user uses the mobile app to either take a photo of a lesion or select an existing image of a lesion. The resulting image is then uploaded to the web server by the mobile app. The web server performs normalization on the uploaded image through resizing, cropping and transformation. The ML model then performs lesion segmentation on the normalized image and subsequently classifies it as either malignant or benign. The results of classification are then relayed back to the web server which returns it to the mobile app which then displays it to the user. The user may then log out.

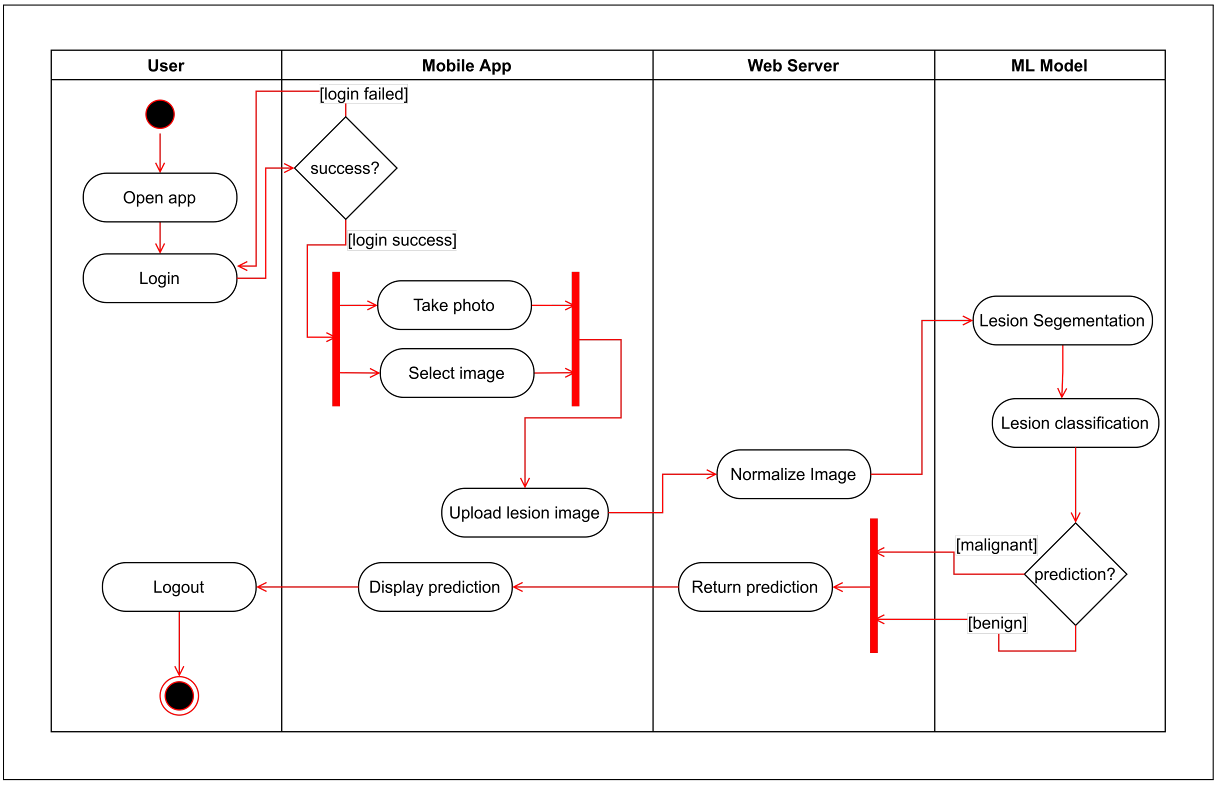


Figure 4.5: Activity diagram

4.3.6 Class Diagram

Figure ?? shows the class diagram for the developed system. The system has three classes namely: User, Lesion and Model classes. The User class has the userId, email properties and the register and login methods. The userId property uniquely identifies a user. The register method creates a new account for a user. The login method authenticates a user.

The Lesion class has lesionId, lesionImage, malignancy, userId, timestamp properties and takeLesionPhoto, selectLesionImage, setLesionImage, setMalignancy, isMalignant, uploadLesion methods. The lesionId property uniquely identifies a lesion, the lesionImage property stores an image file of the lesion, the malignancy property stores a truth value of whether the lesion is malignant or benign and the timestamp property stores the date and time of the creation of a lesion record in the database. The takeLesionPhoto method access the camera on the phone, takes a photo and returns it. The selectLesionImage method allows the user to select an existing lesion image from the file system. The setLesionImage method is used to set the lesionImage property upon capturing of an image or selection from the file system. The isMalignant method uses the uploadLesion method to upload an image of a lesion to the web server requesting for inference on the

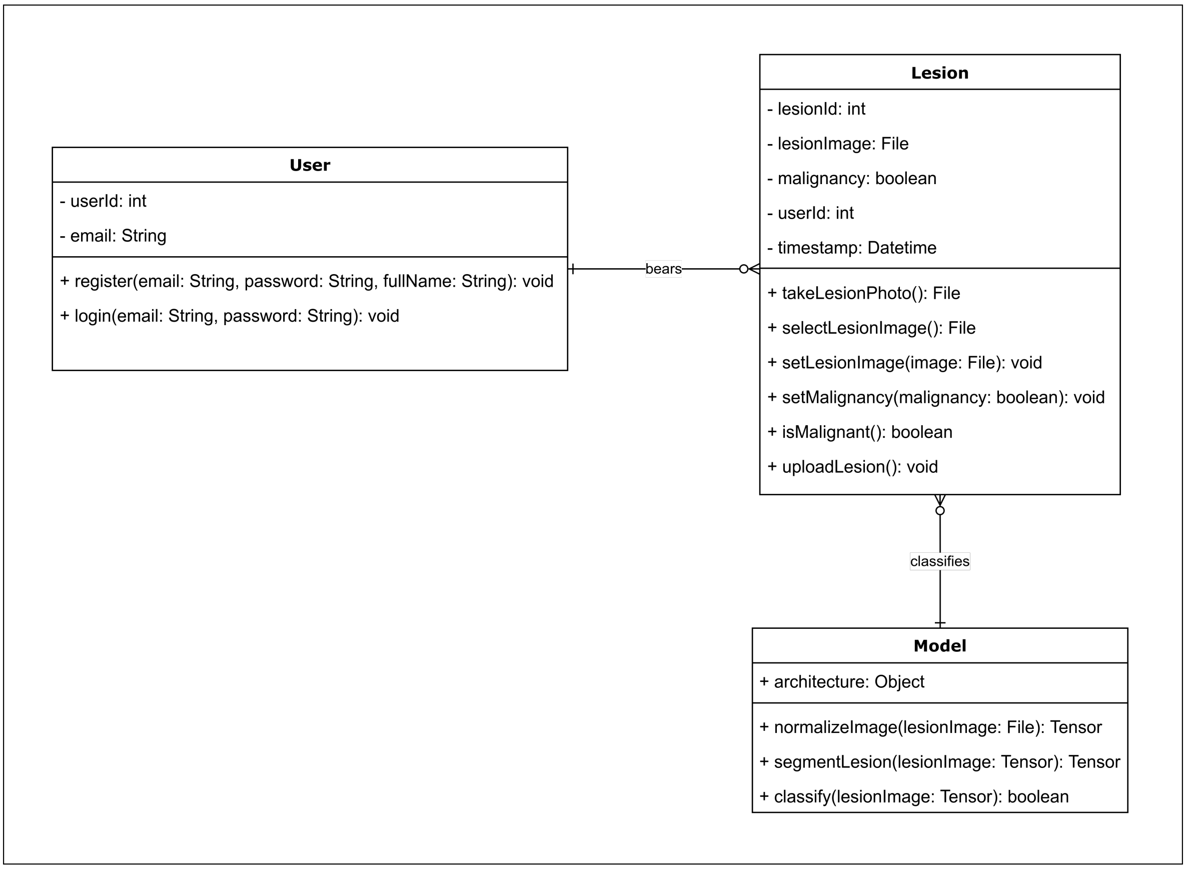

User
- userId: int
- email: String
+ register(email: String, password: String, fullName: String): void
+ login(email: String, password: String): void
bears
Lesion
- lesionId: int
- lesionImage: File
- malignancy: boolean
- userId: int
- timestamp: Datetime
+ takeLesionPhoto(): File
+ selectLesionImage(): File
+ setLesionImage(image: File): void
+ setMalignancy(malignancy: boolean): void
+ isMalignant(): boolean
+ uploadLesion(): void
classifies
Model
+ architecture: Object
+ normalizeImage(lesionImage: File): Tensor
+ segmentLesion(lesionImage: Tensor): Tensor
+ classify(lesionImage: Tensor): boolean

Figure 4.6: Class diagram

lesion, waits on the request and upon the resolution of the request returns a truth value of the inference. The setMalignancy method is a mutator for the malignancy property.

The Model class wraps the architecture of the actual ML model. The Model class provides the normalizeImage method which normalizes an uploaded image in preparation for inference through resizing, cropping and transformation and returns a Tensor representation of the image. The segmentLesion method passes a normalized image tensor to the segmentation model which then returns the resulting tensor. The classify method passes a segmented Tensor lesion image to the ML model, obtains inference and returns it.

4.4 System Design Diagrams

4.4.1 Database Schema

Figure **??** shows the database schema for the developed system. The database consists of two tables namely: User and Lesion tables. The User table has the user_id, email and password attributes. The user_id attribute is the primary key and hence uniquely identifies every single user record. The email and password attribute store the user credentials that is, email and hashed password respectively.

The Lesion table has the lesion_id, user_id, lesion_img_url, lesion_malignancy and lesion_timestamp attributes. The lesion_id is the primary key and hence identifies each lesion record uniquely. The user_id attribute is a foreign key that creates a relationship with the User table such that any lesion belongs to exactly one user. The lesion_img_url attribute store the URL of a lesion's image and the lesion_malignancy nullable attribute stores the inference result of a lesion using either of the two enum values, 0 for benign and 1 for malignant. The lesion_timestamp attribute stores the date and time of the creation of a lesion record.

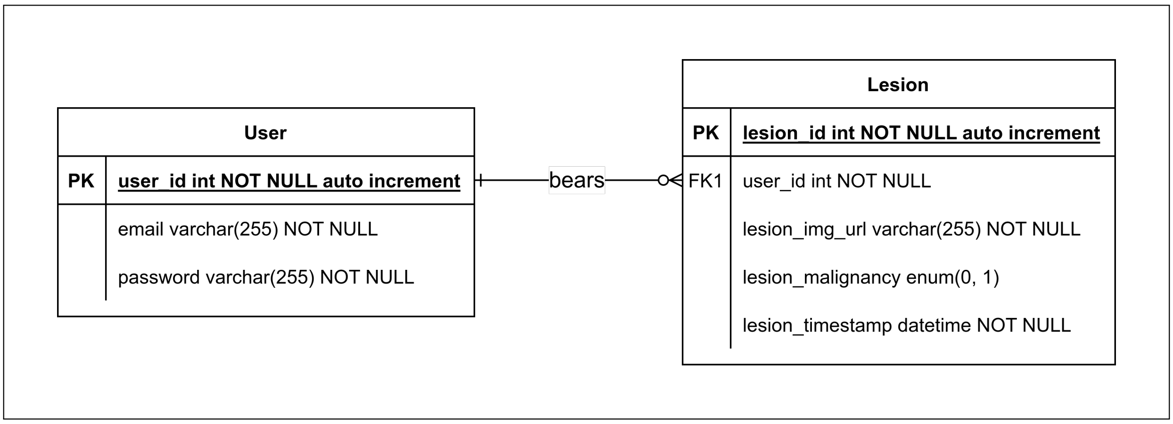

User
PK
user_id int NOT NULL auto increment
email varchar(255) NOT NULL
password varchar(255) NOT NULL
bears
Lesion
PK
lesion_id int NOT NULL auto increment
FK1
user_id int NOT NULL
lesion_img_url varchar(255) NOT NULL
lesion_malignancy enum(0, 1)
lesion_timestamp datetime NOT NULL

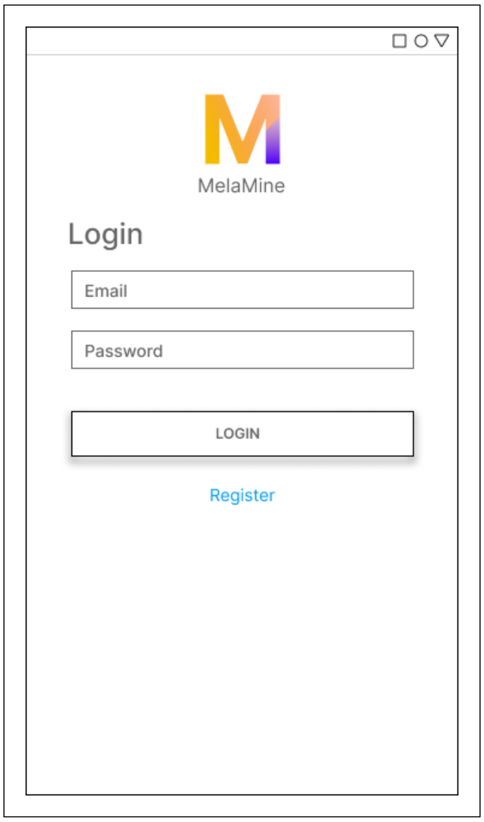
Figure 4.7: Database Schema

4.4.2 Wireframes

4.4.2.1 Authentication Screens

Figure **??** shows the login screen of the developed system. The user is required to enter the correct email and password combination for successful login. The user may also tap on 'Register' in order to create an account. Tapping on 'Register' routes the user to the registration screen displayed in Figure **??**.

On the registration screen, the user is required to enter an email, a new password and its confirmation. The registration is successful if the email address has not been taken and if both the password and its confirmation do match.


MelaMine

Login

Email

Password

LOGIN

Register

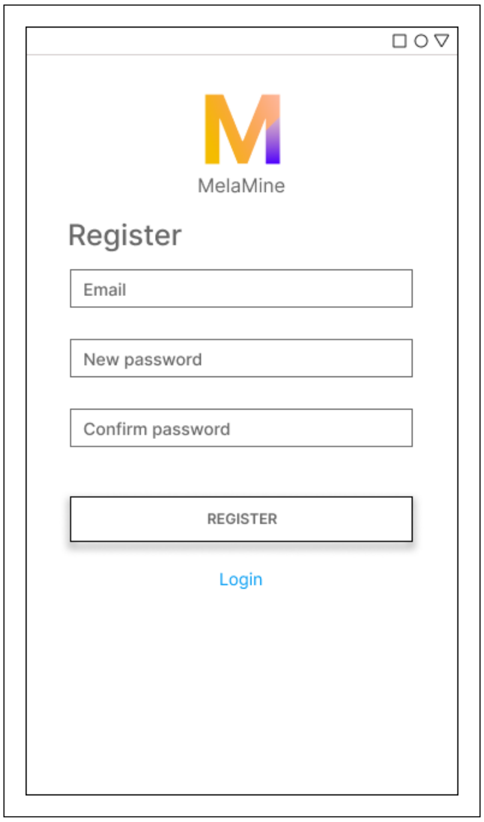
Figure 4.8: Login screen

4.4.2.2 Core Functionality Screens

Upon successful login, the user is routed to the home screen of the mobile application displayed in Figure ??. Tapping on the icon at the top-right of the home screen with an arrow going out of an open rectangle, allows the user to log out.

The user may tap on the hamburger icon shown in Figure ??, upon which the user is routed to the screen in Figure ?? where a list of predictions made on lesions in the past is displayed. The predictions are sorted by date in descending order. Selecting any of the listed predictions shows the user the image of the lesion and the prediction made on the image as in either Figure ?? or Figure ??.

On the home screen, the user can tap on the big round button with a camera icon in order to take a photo of a lesion. Alternatively, the user can tap on 'SELECT IMAGE' to select an image from the file system instead. If the user selects the camera icon button they are routed to the screen shown in Figure ?? where a live stream for the currently active camera is displayed waiting for the user to take a photo.


MelaMine

Register

Email

New password

Confirm password

REGISTER

Login

Figure 4.9: Registration screen

Upon taking a photo or selecting an image from the file system the user is routed to the image upload screen shown in Figure ??. The user can avoid uploading the currently displayed image by tapping on the 'X' button. Otherwise, the user can upload the image for inference by tapping the button with the "send" icon upon which the user is presented with the screen shown in Figure ?? where the user is instructed to wait as the image is uploaded and subsequently undergoing inferencing.

Once inference results have been returned by the server the user is routed to the screen in Figure ?? if the prediction is positive, that is, the lesion has been predicted as likely to be malignant melanoma. Conversely, if the prediction is negative, that is, the lesion has been predicted as unlikely to be melanoma, the user is routed to the screen in Figure ??.

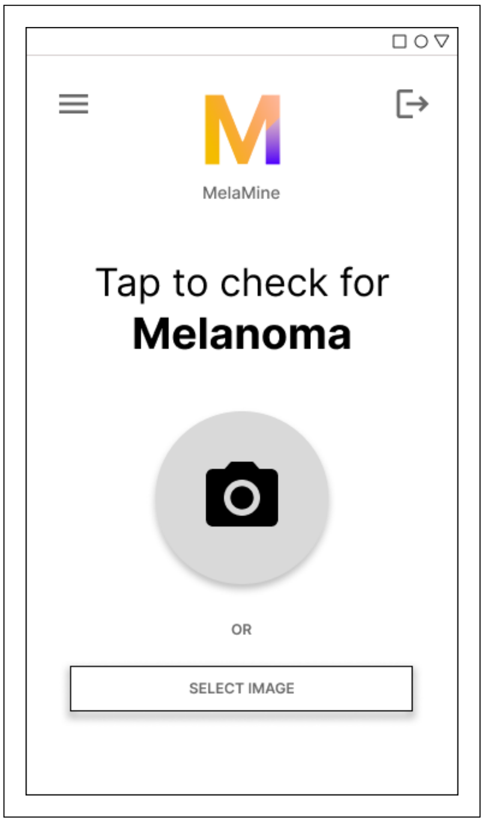


Figure 4.10: Home screen

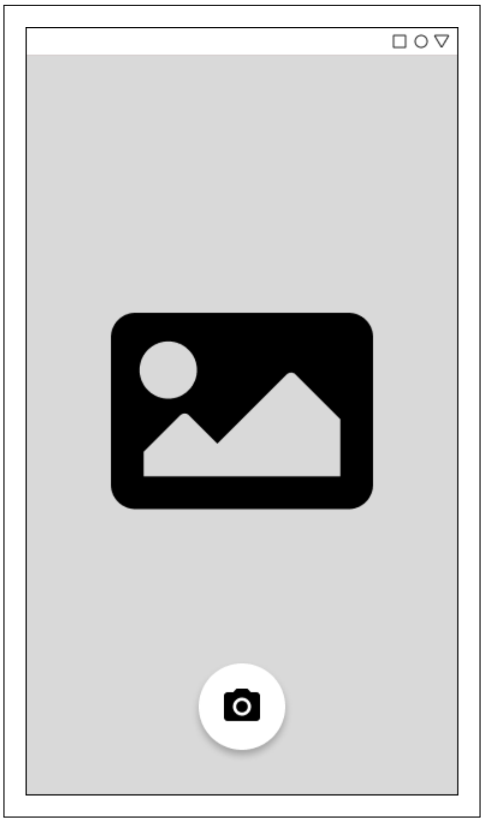


Figure 4.11: Image capture screen

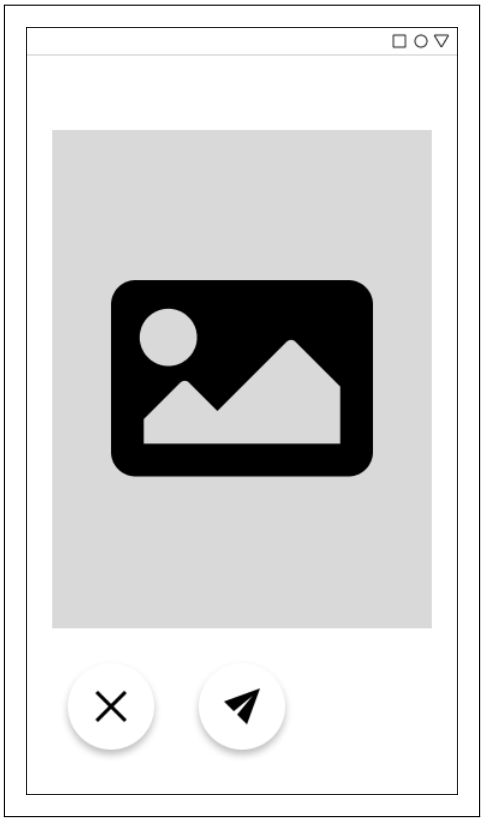


Figure 4.12: Image upload screen

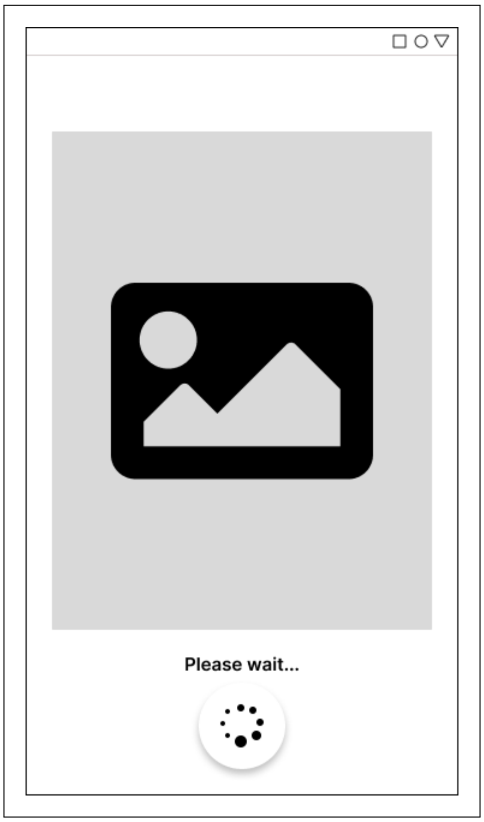
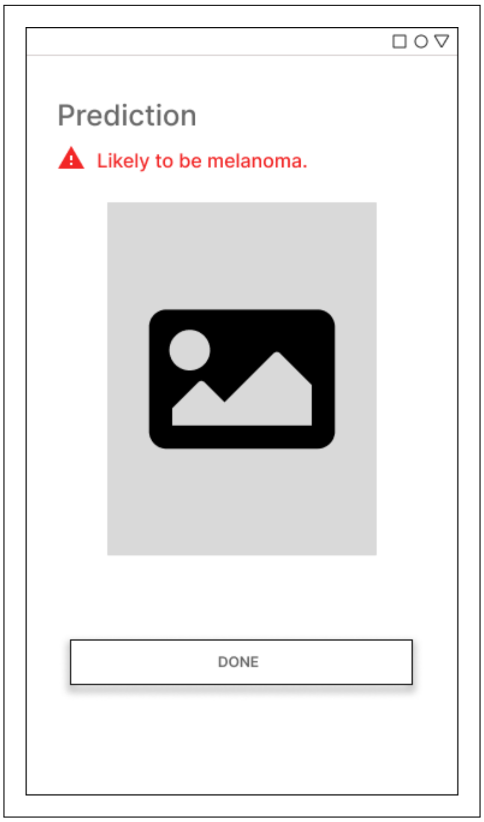
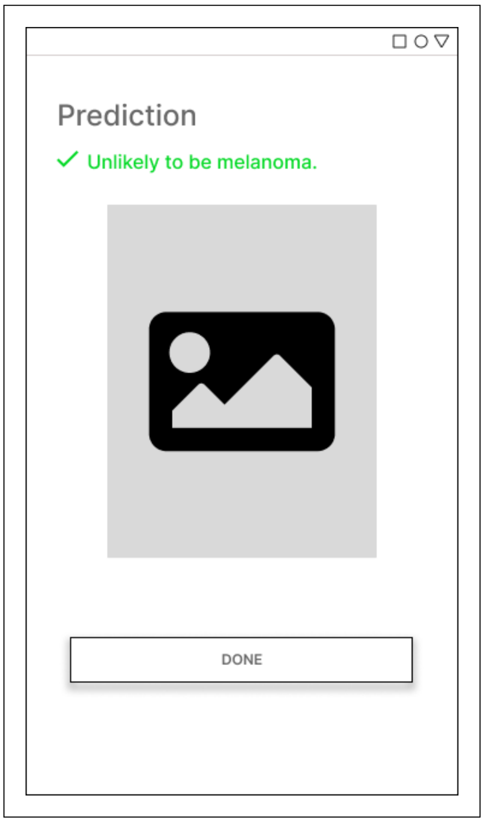


Figure 4.13: Waiting-for-prediction screen


Prediction

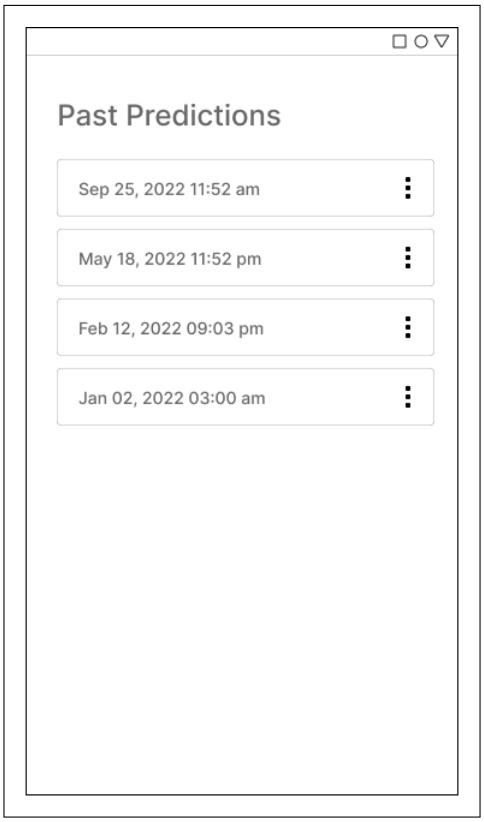
⚠ Likely to be melanoma.

Figure 4.14: Positive prediction display screen


Prediction

✓ Unlikely to be melanoma.

Figure 4.15: Negative prediction display screen



Past Predictions

Sep 25, 2022 11:52 am

May 18, 2022 11:52 pm

Feb 12, 2022 09:03 pm

Jan 02, 2022 03:00 am

Figure 4.16: Past predictions display screen

4.4.3 System Architecture

Figure ?? shows the system architecture diagram for the developed system. The developed system consists of a thin client React Native mobile app, a Flask web server hosting a PyTorch environment where the ML model operates from and a MySQL database. The mobile app allows the user to capture an image or select an image from the file system and upload it to the Flask web server for inference. The Flask web server passes the uploaded image to the ML model and obtains inference results which it then relays back to the mobile app. The Flask web server stores user data and prediction history in the SQL server and mediates data retrieval from the database by the mobile app.

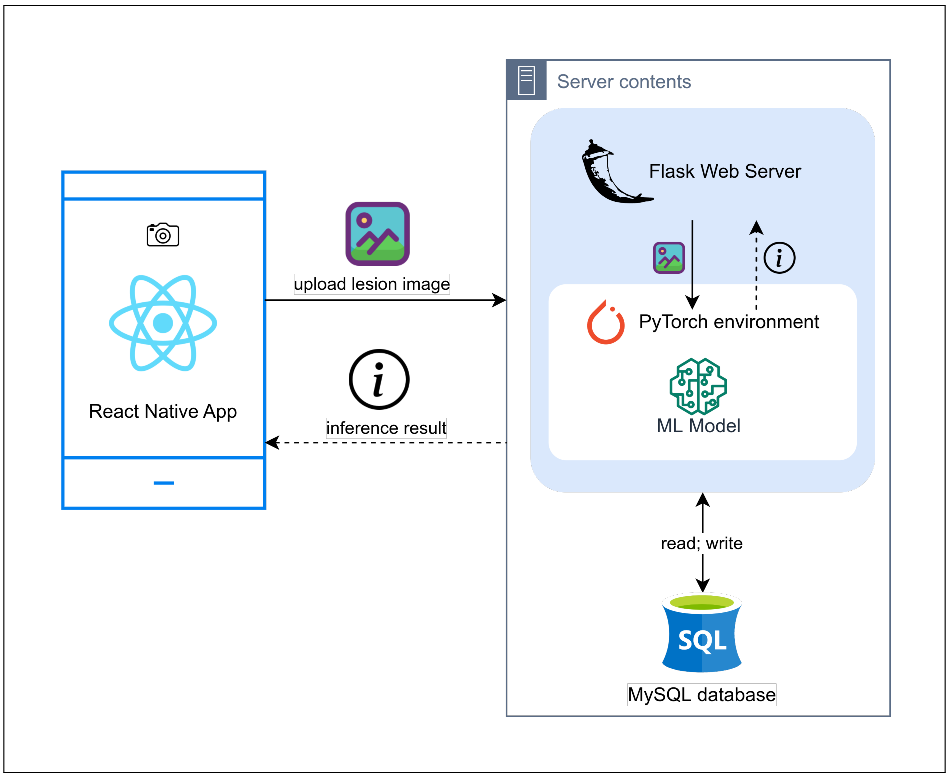


Figure 4.17: System architecture diagram

Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter discusses how the developed solution was implemented and tested. The hardware and software specifications of the implementation environment are discussed, followed by the description of the dataset used. Moreover, the description of training and the description of testing is discussed. The chapter concludes by discussing the testing paradigm used and the testing results.

5.2 Description of the Implementation Environment

5.2.1 Hardware Specifications

The minimum hardware specifications of the developed solution are as shown in Table **??** below. The developed solution is composed of a minimum of two devices interacting which include: a web server and a mobile device.

Device	**Hardware**	**Description and Justification**
Web server	CPU: at least 1 gigahertz clock speed	This is the minimum requirement for installing Ubuntu Server Edition operating system. Moreover, the machine learning model uses the CPU to perform inference.
	2 Gigabytes of RAM	This ensures that the Ubuntu Server OS runs smoothly and can also effectively host a web server capable of performing inference for image classification.

	4 Gigabytes of disk space	Ubuntu Server Edition OS needs a minimum of 2.5 Gigabytes of disk space and the remaining 1.5 Gigabytes is space for installing software packages, storing web server code and storing image uploads.
Mobile device	CPU: Quad Core 1.3 gigahertz clock speed	Required to run Android 5 OS and higher.
	1 Gigabyte of RAM	This is the minimum requirement to run Android 5 or higher.
	100 Megabytes of disk space	This storage space is required to install the application and to be able to take photos and store them.

Table 5.1: Table showing hardware specifications

5.2.2 Software Specifications

The developed solution is composed of the following software components: a mobile application, and a RESTful API. The software specifications for the above software components are as shown in Table **??** below.

Software	**Description and Justification**
Android 5+ and iOS 13+	Since, the mobile application for the developed solution is built using Expo, which is a React Native framework, the mobile devices need to run Android 5+ and iOS 13+.
Expo SDK 46	The mobile application was written using Expo SDK 46, hence it is required for one to run the project and build application bundles for distribution on Play Store and App Store. Expo was used because of the convenience of being ab;e to write cross-platform applications in Javascript.
Flask	The RESTful API was written using Flask, a micro web framework written in Python. Flask was used because of its simplicity and also because the machine learning model could easily be run from the same API.
NPM(Node Package Manager)	Node package manager were used to install, update and manage dependencies for the development of the mobile application.
PIP	PIP was used to install and manage software packages for the development of the RESTful API.
PyTorch	PyTorch was used as the machine learning framework. Its torch and torchvision packages were used.

Table 5.2: Table showing software specifications

5.3 Description of the Dataset

To train the model used in the developed solution and five different but related datasets were combined and used. The datasets used are shown in Table ??. The datasets consisted of images of two classes: malignant melanoma and benign nevus. The number of images for each class is shown in Table ?? below. All the datasets were downloaded from where they were hosted on the internet with all of them being used and cited in published research.

Dataset	**Publisher**	**No. of images**
MED-NODE	?	170
PAD-UFES-20	?	296
7-point	?	827
DermIS	?	69
DermQuest	Galderma S.A.	137
	Total	**1499**

Table 5.3: Table showing list of datasets to be used

Before the ensemble dataset could be used, it had to be pre-processed. Using the metadata accompanying the datasets, Talend, an ETL tool, was used to combine the datasets into a uniform and orderly whole. The PAD-UFES-20 dataset (?) and the 7-point dataset (?) included images for other types of skin cancer which were not used to train the model. Hence, Talend was used to extract the images corresponding to malignant melanoma and benign naevi from those datasets.

Class	**Count**	**Percentage**
Malignant Melanoma	493	32.89%
Benign Nevus(naevi)	1006	67.11%

Table 5.4: Table showing number of images for each class

The combined dataset was split into train, test and validation subsets as shown in Table ?? below. Since the five datasets were drawn from slightly different distributions and owing to the class imbalance as can be seen in Table ??, random stratified sampling was used to split the combined dataset such that all the different distributions and the two classes were represented proportionately in the train, validation and evaluation sets.

Subset	**Percentage**
Train Set	70%
Validation Set	15%
Evaluation Set	15%

Table 5.5: Table showing training, validation and evaluation sets split

Figure ?? below shows a sample of images from the ensemble dataset where the images labelled 'melanoma' correspond to malignant skin lesions while those labelled 'nevus' correspond to benign skin lesions.

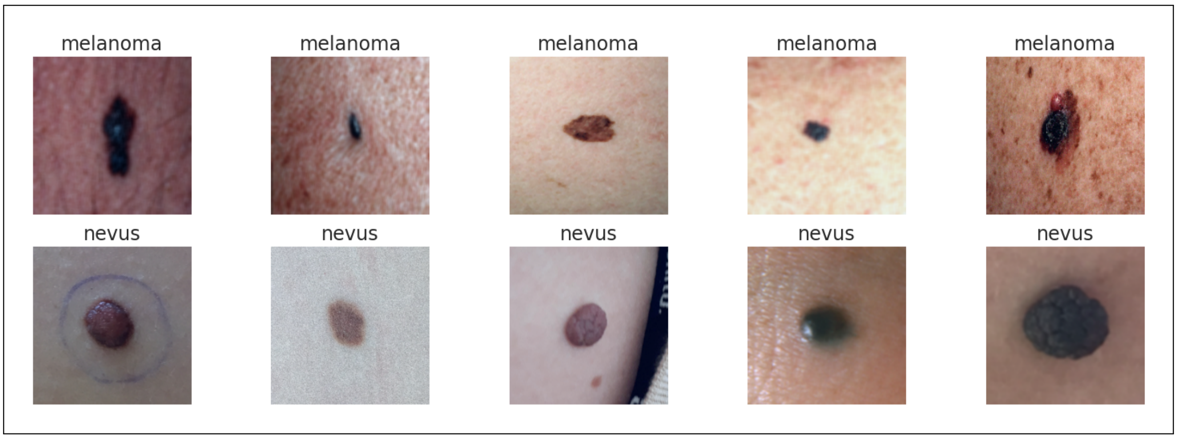


Figure 5.1: Figure showing sample images from the ensemble dataset

5.4 Description of Training

This section discusses how the segmentation and classification models were trained and various experiments conducted during model training which include: exploring learning rate schedulers, investigating the effect of data augmentation on performance, investigating the effect of image segmentation on performance and exploring transfer learning with different pre-trained models. All training experiments were done using Pytorch and hosted on Google Colaboratory. In order to ensure the reproducibility of the experiments, random number generators in the execution environment were seeded.

5.4.1 Training of Classification Models

The loss function used to train classification models was the Cross-Entropy Loss function whose equation is shown below, where *log* is natural logarithm, $p_{o,c}$ is the predicted probability for a class c, $y_{o,c}$ is the binary target label for a class c and M is the number of classes which in this case is 2.

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{1}$$

The optimizer used was Adam which in full form is Adaptive Moment Estimation. An early stopping technique was used to prevent overfitting of the model and reduce training time. All models used were initialized with pre-trained weights. Only the classification layers of the neural network models were replaced in order to carry out binary classification as opposed to multiclass classification which they were trained on using the Imagenet dataset. For this reason, the models' classifier parameters were fine-tuned for several epochs while freezing the rest of the network parameters to ensure the stability of the model.

Before feeding images to the models, images were read from storage to memory and resized to $256 * 256$ pixels, centre-cropped to $224 * 224$, standardised using mean and standard deviation values for the red, green and blue channels shown in Table **??** below and finally converted into Pytorch tensors. Checkpoints used were selected based on the loss evaluated on the validation set where checkpoints with the least loss for each of the pre-trained models were chosen.

Channel	**Mean**	**Standard deviation**
Red	0.485	0.229
Green	0.456	0.224
Blue	0.406	0.224

Table 5.6: Table showing image standardization values for RGB channels

5.4.2 Training U-Net Segmentation Model

According to the study conducted by **?**, image segmentation leads to better performance in the classification of melanoma skin lesions. This study was the basis of the incorporation of image segmentation during model training. The image segmentation model used was U-Net which was implemented and trained using the Dermis and DermQuest datasets which included lesion images and ground truth segmentation masks. The two datasets were combined and splits into train and validation sets comprising 80% and 20% of the total number of images respectively. In order to improve segmentation performance and reduce overfitting data augmentation was applied comprising of probabilistic transformations of rotation, horizontal flip and vertical flip.

The loss function used was the Cross-Entropy Loss function depicted in Equation 5.1. Before feeding images to the model images were read from storage to memory and resized to $350 * 350$ pixels, standardised using a mean of 0 and standard deviation of 1 for the red, green and blue channels and finally converted into Pytorch tensors.

5.4.3 Exploring Learning Rate Schedulers

Various learning rate schedulers were used including Cosine Annealing with Warm Restarts, One Cycle learning rate scheduler and the Step Learning rate scheduler. Upon training and evaluation, the One Cycle learning rate scheduler was chosen since it had the best evaluation metrics performance as shown in Table **??** below (highest scores per metric are in bold).

Learning rate scheduler	**Accuracy**	**F1-Measure**	**Sensitivity**	**Specificity**

Cosine Annealing with Warm Restarts	0.7753	0.7753	0.8410	0.64473
One Cycle	**0.8326**	**0.8716**	0.8543	**0.7894**
Step	0.8061	0.8061	**0.9006**	0.6184

Table 5.7: Table showing testing results for various learning rate schedulers

5.4.4 Effect of Data Augmentation

According to **?**, data augmentation is a technique of expanding the size of a dataset by applying several random transformations to original images. The Albumenation library developed by **?** was used where a series of transformations were applied to the images with a given probability of being applied. Applying these transformations had the effect of increasing the number of samples in the dataset without actually saving new images to storage since the transformations are applied with a given probability, hence different images with different transformations were fed to the model across the epochs. The transformation applied and listed in Table **??** below.

Transformation	Description	Probability
Random rotate 90 degrees	Randomly rotate the input by 90 degrees zero or more times.	0.5
Flip	Flip the input either horizontally, vertically or both.	0.5

Transpose	Transpose the input by swapping rows and columns.	0.5
Additive Gaussian Noise	Add Gaussian noise	0.5
Blur	Blur the input image using a random-sized kernel.	0.05
Motion blur	Apply motion blur to the input image.	0.1
Median blur	Blur the input image using a median filter	0.05
Random brightness and contrast	Randomly change the brightness and contrast of the input image.	0.3
Hue, saturation and value	Randomly change the hue, saturation and value of the input image.	0.3
Gaussian Blur	Blur the input image using a Gaussian filter with a random kernel size.	0.5

Table 5.8: Table listing data augmentation transformations

The use of data augmentation led to significantly better performance as compared to training without data augmentation as shown in Table ?? using the Resnet34 model. The increase in performance as a result of data augmentation can be explained by Figure ?? which depicts the graphs of validation loss against epochs and accuracy against epochs. The trend of the curves suggests more resistance to overfitting compared to the same graphs for training without data augmentation shown in Figure ?? (highest scores per metric are in bold).

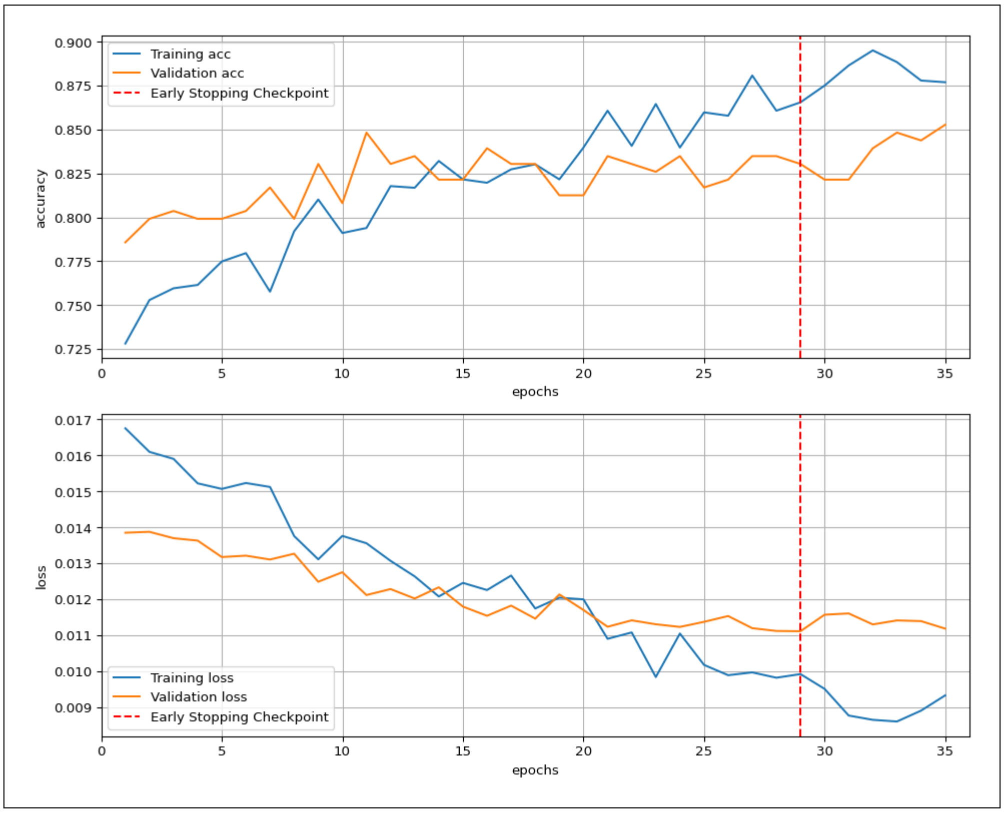


Figure 5.2: Graphs of loss and accuracy against epochs(using data augmentation)

	Accuracy	**F1-Measure**	**Sensitivity**	**Specificity**
Without augmentation	0.7929	0.7929	**0.8741**	0.6316
With augmentation	**0.8326**	**0.8716**	0.8543	**0.7894**

Table 5.9: Table showing testing results of data augmentation experiments

5.4.5 Effect of Image Segmentation

Using a pre-trained ResNet34 model, performance was compared when training the model using segmented lesion images and non-segmented lesion images. As shown in Table **??** below, segmentation led to a poorer performance as compared to training without image segmentation across all evaluation metrics (highest scores per metric are in bold). As a result, the developed solution does not include image segmentation. These results affirm the findings of **?**, who asserts that image segmentation is a non-trivial task which risks leaving out important features hence negatively impacting classification performance.

	Accuracy	**F1-Measure**	**Sensitivity**	**Specificity**
With segmentation	0.7577	0.7577	0.8476	0.5789
Without segmentation	**0.8326**	**0.8716**	**0.8543**	**0.7894**

Table 5.10: Table showing testing results of image segmentation experiments

5.4.6 Transfer Learning with Different Pretrained Models

Research has shown that transfer learning significantly increases model accuracy while reducing training time and computing resources. The pre-trained models used included ResNet34, ResNet50, ResNet152, InceptionV3, EfficientNetV2 and VisionTransformer. All of the models were initialized with pre-trained weights trained on Imagenet (**?**) which is a large-scale hierarchical image database commonly used as a computer vision benchmark dataset. The performance metrics of the trained models are discussed in Section 5.5.

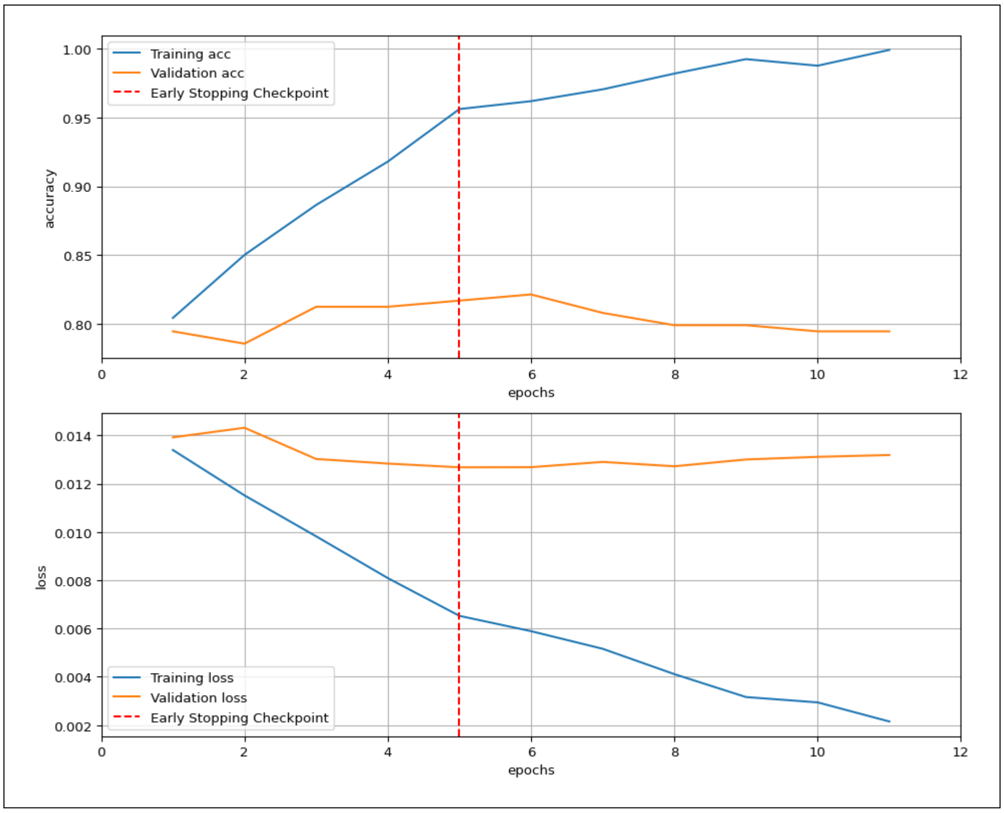


Figure 5.3: Graphs of loss and accuracy against epochs(without data augmentation)

5.5 Description of Testing and Evaluation

In order to test and evaluate the models trained the validation set and evaluation set respectively (listed in Table **??**), were used.

5.5.1 Evaluation Metrics Used

Evaluation metrics used include accuracy, sensitivity, specificity, F1-measure, and dice score which are expounded on in Table **??** below.

Metric	**Description**	**Equation**
Sensitivity	Measures model's ability to predict true positives as positive.	$= \frac{TP}{TP+FN}(2)$
Specificity	Measures model's ability to predict true negatives as negative	$\frac{TN}{FP+TN}(3)$

F1-Measure	Harmonic mean of precision and recall	$= \frac{2*TP}{2*TP+FP+FN}(4)$
Accuracy	Ratio of total correctly classified observation to the total number of observations	$= \frac{TP+TN}{TP+TN+FP+FN}(5)$
Dice Score	Is a metric used to gauge image segmentation performance which is the ratio between twice the area of overlap between the region of an image and its segmentation mask, to the total number of pixels in an image and its segmentation mask	$= \frac{2\|U\cap V\|}{\|U\cup V\|}(6)$

Table 5.11: Table showing evaluation metrics used

5.5.2 Experimental Results

The U-Net model was trained to segment skin lesions using the DermIS and DermQuest datasets which had ground truth image segmentation marks. The U-Net model obtained an accuracy of 0.9232 and a dice score of 0.7414 on its validation set.

Table **??** show the performance metrics of the classification models resulting from training (the highest scores per metric are in bold). The models were trained using the one-cycle learning rate scheduler, with data augmentation applied and without image segmentation.

Model	**Accuracy**	**F1-Measure**	**Sensitivity**	**Specificity**	**Average**
Resnet34	**0.8326**	0.8716	0.8543	0.7894	**0.8370**
Resnet50	0.7974	0.8392	0.7947	**0.8026**	0.8085

Resnet152	**0.8326**	0.8701	0.8874	0.6974	0.8197
InceptionV3	0.8194	0.8707	**0.9139**	0.6316	0.8089
EfficientNetV2	0.8194	0.8638	0.8609	0.7368	0.8202
VisionTransformer	0.8282	**0.8762**	0.9139	0.6579	0.8190

Table 5.12: Table showing testing results for various pre-trained models

Confusion matrices for the Resnet34, Resnet50, Resnet152, InceptionV3, EfficientNetV2 and VisionTransfomer are displayed in Appendix I. From the experimental results, it turned out that the best model is Resnet34 since it has the highest average score which was calculated as the arithmetic mean of accuracy, precision, sensitivity and specificity. For this reason, the trained Resnet34 was used for classification in the developed solution.

5.6 Testing Paradigm

A hybrid test paradigm was used for the developed solution comprising black box testing, white box testing and unit testing.

5.6.1 Black Box Testing

Black box testing is a software validation technique where the correctness and appropriateness of software are evaluated based on its output in response to an input given and conditions it is subjected to without investigating the internal workings of the software being tested **?**. Black box comprised the majority of the test coverage and it was used to validate the functioning of the RESTful API and the mobile application.

5.6.2 White Box Testing

White box testing is a software validation technique where the internal workings of implementation source code are investigated to identify bugs in software **?**. white box testing was used throughout the development process to ensure that the source code actually did what it was intended to do. White box testing to ensure the correctness of the source code for the RESTful API and the mobile application.

5.7 Testing Results

Table ?? shows the results of black box testing carried out on the Flask RESTful API.

Test Case	Description	Test Data	Result	Test Verdict
TC01: Obtaining prediction for a lesion image	Attempt made to upload image to API, get inference and store record in database	POST HTTP request to http://¡ip address¿:port/api/lesions/¡id¿, where ¡id¿ is the user's id	Inference result returned for image and new record created for it.	Pass
TC02: Retrieving list of past predictions	Attempt made to retrieve list of past predictions for a given user	GET HTTP request to http://¡ip address¿:port/api/lesions/¡id¿, where ¡id¿ is the user id of the currently logged in user.	JSON array of existing predictions in the lesions table is returned.	Pass
TC03: Deleting past prediction	Attempt made to delete past prediction from database.	DELETE HTTP request to http://¡ip address¿:port/api/lesions/¡id¿, where ¡id¿ is an id for a well-known past prediction	Message for successful deletion returned and record deleted from lesions table	Pass

Table 5.13: Table showing testing results for the RESTful API

Table ?? below shows the results of the testing carried out on the developed mobile app.

Test Case	Description	Test Data	Result	Test Verdict
TC04: Taking lesion photo	Attempt made to take a photo of skin lesion	Tap button with camera icon then take photo	User is redirected to the camera app and photo taken and saved to device storage and displayed to user in the app.	Pass
TC05: Selecting lesion photo	Attempt made to select existing lesion photo from storage	Tapping the ”Select image” button	User is directed to a file picker allowing picking of images only and selected image displayed in the app.	Pass
TC06: Uploading lesion photo for inference	Attempt made to upload and obtain inference for a lesion photo	Tap camera icon button, then take lesion photo, tap the check icon and wait	User is directed to camera app, takes photo, photo is successfully uploaded and inference results displayed.	Pass

TC07: Viewing past predictions	Attempt made to obtain a prediction and confirm its presence in history of past predictions	Tap camera icon button, take a photo, upload photo and get prediction. Then press back and click on history icon button	After obtaining prediction, a record of the new prediction was seen to appear under the list of past predictions.	Pass
TC08: Deleting past prediction	Attempt made to delete past prediction entry	Tap history icon and press delete bin icon of an entry	On pressing the delete bin icon for an entry, the entry is removed from list of past predictions.	Pass

Table 5.14: Table showing testing results for the mobile application

Chapter 6: Conclusions, Recommendations and Future Works

6.1 Conclusions

Having obtained high-performance metric scores for the Resnet34 model (accuracy of 0.8326, F1-Measure of 0.8716, sensitivity of 0.8543 and specificity of 0.7894) for the classification of melanoma skin lesions, it can be concluded that indeed the developed solution can reliably be used to support the diagnosis of melanoma where there are few diagnostic specialists and increase chances of early detection of melanoma. Furthermore, this study showed that the segmentation of skin lesion images before classification negatively impacts classification performance. Additionally, it was concluded that data augmentation significantly increases model performance by reducing overfitting.

6.2 Recommendations

It is recommended that for training various deep learning models, the reproducibility of the results should be taken into account in order to ensure consistent results and conclusions. This is because the training of deep learning models involves the use of random number generators hence if random number generators are not seeded inconsistent results will be obtained.

6.3 Future Works

In view of the limitation on the size of melanoma skin lesion datasets, future works in this area should focus on the collection of more non-dermoscopic images to create larger datasets in order to increase confidence in the performance of deep learning models.

Bibliography

Abbasi, N. R., Shaw, H. M., Rigel, D. S., Friedman, R. J., McCarthy, W. H., Osman, I., Kopf, A. W., and Polsky, D. (2004). Early Diagnosis of Cutaneous MelanomaRevisiting the ABCD Criteria. *JAMA*, 292(22):2771–2776.

Abdelhalim, I. S. A., Mohamed, M. F., and Mahdy, Y. B. (2021). Data augmentation for skin lesion using self-attention based progressive generative adversarial network. *Expert Systems with Applications*, 165:113922.

Argenziano, G., Albertini, G., Castagnetti, F., De Pace, B., Di Lernia, V., Longo, C., Pellacani, G., Piana, S., Ricci, C., and Zalaudek, I. (2012). Early diagnosis of melanoma: what is the impact of dermoscopy? *Dermatologic Therapy*, 25(5):403–409.

Argenziano, G., Fabbrocini, G., Carli, P., De Giorgi, V., Sammarco, E., and Delfino, M. (1998). Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions: comparison of the abcd rule of dermatoscopy and a new 7-point checklist based on pattern analysis. *Archives of dermatology*, 134(12):1563–1570.

Bhattacharya, A., Young, A., Wong, A., Stalling, S., Wei, M., and Hadley, D. (2017). Precision diagnosis of melanoma and other skin lesions from digital images. *AMIA Summits on Translational Science Proceedings*, 2017:220.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. *Information*, 11(2):125.

Codella, N. C., Nguyen, Q.-B., Pankanti, S., Gutman, D. A., Helba, B., Halpern, A. C., and Smith, J. R. (2017). Deep learning ensembles for melanoma recognition in dermoscopy images. *IBM Journal of Research and Development*, 61(4/5):5–1.

Corneli, P., Zalaudek, I., Magaton Rizzi, G., and di Meo, N. (2018). Improving the early diagnosis of early nodular melanoma: can we do better? *Expert Review of Anticancer Therapy*, 18(10):1007–1012.

Davis, L. E., Shalin, S. C., and Tackett, A. J. (2019). Current state of melanoma diagnosis and treatment. *Cancer biology & therapy*, 20(11):1366–1379.

de Carvalho, T. M., Noels, E., Wakkee, M., Udrea, A., and Nijsten, T. (2019). Development of smartphone apps for skin cancer risk assessment: progress and promise. *JMIR Dermatology*, 2(1):e13376.

de Lima, L. M. and Krohling, R. A. (2022). Exploring advances in transformers and cnn for skin lesion diagnosis on small datasets. *arXiv preprint arXiv:2205.15442*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Di Leo, C., Bevilacqua, V., Ballerini, L., Fisher, R., Aldridge, B., and Rees, J. (2015). Hierarchical classification of ten skin lesion classes. In *Proc. SICSA Dundee Medical Image Analysis Workshop*.

Di Leo, G., Paolillo, A., Sommella, P., and Fabbrocini, G. (2010). Automatic diagnosis of melanoma: a software system based on the 7-point check-list. In *2010 43rd Hawaii international conference on system sciences*, pages 1–10. IEEE.

Divito, S. J. and Ferris, L. K. (2010). Advances and short comings in the early diagnosis of melanoma. *Melanoma Research*, 20(6):450–458.

Domingues, B., Lopes, J. M., Soares, P., and Pópulo, H. (2018). Melanoma treatment in review. *ImmunoTargets and therapy*, 7:35.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dzwierzynski, W. W. (2013). Managing malignant melanoma. *Plastic and reconstructive surgery*, 132(3):446e–460e.

Engasser, H. C. and Warshaw, E. M. (2010). Dermatoscopy use by us dermatologists: a cross-sectional survey. *Journal of the American Academy of Dermatology*, 63(3):412–419.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118.

Fink, C. and Haenssle, H. (2017). Non-invasive tools for the diagnosis of cutaneous melanoma. *Skin Research and Technology*, 23(3):261–271.

Giavina-Bianchi, M., Santos, A. P., and Cordioli, E. (2020). Teledermatology reduces dermatology referrals and improves access to specialists. *EClinicalMedicine*, 29:100641.

Giotis, I., Bunte, K., Petkov, N., and Biehl, M. (2013). Adaptive matrices and filters for color texture classification. *Journal of Mathematical Imaging and Vision*, 47(1):79–92.

Giotis, I., Molders, N., Land, S., Biehl, M., Jonkman, M. F., and Petkov, N. (2015). Med-node: A computer-assisted melanoma diagnosis system using non-dermoscopic images. *Expert systems with applications*, 42(19):6578–6585.

Glinz, M. (2007). On non-functional requirements. In *15th IEEE international requirements engineering conference (RE 2007)*, pages 21–26. IEEE.

Grant-Kels, J. M., Bason, E. T., and Grin, C. M. (1999). The misdiagnosis of malignant melanoma. *Journal of the American Academy of Dermatology*, 40(4):539–548.

Han, S. S., Kim, M. S., Lim, W., Park, G. H., Park, I., and Chang, S. E. (2018). Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. *Journal of Investigative Dermatology*, 138(7):1529–1538.

He, H. (2019). The state of machine learning frameworks in 2019. *The Gradient*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Heidelburg, U. and Erlangen, U. (2022). Dermatology information system. `https://www.dermis.net/dermisroot/en/home/index.htm`. Accessed: 12-07-2022.

Jafari, M. H., Nasr-Esfahani, E., Karimi, N., Soroushmehr, S., Samavi, S., and Najarian, K. (2017). Extraction of skin lesions from non-dermoscopic images for surgical excision of melanoma. *International journal of computer assisted radiology and surgery*, 12(6):1021–1030.

Jafari, M. H., Samavi, S., Soroushmehr, S. M. R., Mohaghegh, H., Karimi, N., and Najarian, K. (2016). Set of descriptors for skin cancer diagnosis using non-dermoscopic color images. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2638–2642.

Jaworek-Korjakowska, J. and Kłeczek, P. (2016). Automatic classification of specific melanocytic lesions using artificial intelligence. *BioMed research international*, 2016.

Jaworek-Korjakowska, J. and Kleczek, P. (2018). Eskin: study on the smartphone application for early detection of malignant melanoma. *Wireless Communications and Mobile Computing*, 2018.

Jones, O., Jurascheck, L., Van Melle, M., Hickman, S., Burrows, N., Hall, P., Emery, J., and Walter, F. (2019). Dermoscopy for melanoma detection and triage in primary care: a systematic review. *BMJ open*, 9(8):e027529.

Kabir, M. M., Mehera, P., Saha, A. K., Mridha, M. F., Hamid, M. A., and Monowar, M. M. (2021). Cnn-nsvm architecture for skin lesion classification using non-dermoscopic digital image. In *2021 Joint 10th International Conference on Informatics, Electronics and Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision and Pattern Recognition (icIVPR)*, pages 1–7.

Kassianos, A. P., Emery, J., Murchie, P., and Walter, F. M. (2015). Smartphone applications for melanoma detection by community, patient and generalist clinician users: a review. *British Journal of Dermatology*, 172(6):1507–1518.

Kawahara, J., BenTaieb, A., and Hamarneh, G. (2016). Deep features to classify skin lesions. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 1397–1400.

Kawahara, J., Daneshvar, S., Argenziano, G., and Hamarneh, G. (2019). Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE Journal of Biomedical and Health Informatics*, 23(2):538–546.

Lee, J. B. and Hirokawa, D. (2010). Dermatoscopy: Facts and controversies. *Clinics in dermatology*, 28(3):303–310.

Marghoob, A. A., Changchien, L., DeFazio, J., Dessio, W. C., Malvehy, J., Zalaudek, I., Halpern, A. C., and Scope, A. (2009). The most common challenges in melanoma diagnosis and how to avoid them. *Australasian Journal of Dermatology*, 50(1):1–13.

Mosam, A. and Todd, G. (2021). Dermatology training in africa: successes and challenges. *Dermatologic Clinics*, 39(1):57–71.

Nasr-Esfahani, E., Samavi, S., Karimi, N., Soroushmehr, S., Jafari, M., Ward, K., and Najarian, K. (2016). Melanoma detection by analysis of clinical images using convolutional neural network. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1373–1376.

Nerson, J.-M. (1992). Applying object-oriented analysis and design. *Communications of the ACM*, 35(9):63–74.

Nidhra, S. and Dondeti, J. (2012). Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications (IJESA)*, 2(2):29–50.

Okur, E. and Turkan, M. (2018). A survey on automated melanoma detection. *Engineering Applications of Artificial Intelligence*, 73:50–67.

Pacheco, A. G. and Krohling, R. A. (2020). The impact of patient clinical information on automated skin cancer detection. *Computers in Biology and Medicine*, 116:103545.

Pehamberger, H., Binder, M., Steiner, A., and Wolff, K. (1993). In vivo epiluminescence microscopy: Improvement of early diagnosis of melanoma. *Journal of Investigative Dermatology*, 100(3, Supplement):S356–S362.

Pérez, E., Reyes, O., and Ventura, S. (2021). Convolutional neural networks for the automatic diagnosis of melanoma: An extensive experimental study. *Medical image analysis*, 67:101858.

Pérez, E. and Ventura, S. (2021). An ensemble-based convolutional neural network model powered by a genetic algorithm for melanoma diagnosis. *Neural Computing and Applications*, pages 1–20.

Psaty, E. L. and Halpern, A. C. (2009). Current and emerging technologies in melanoma diagnosis: the state of the art. *Clinics in dermatology*, 27(1):35–45.

Raschka, S. (2015). *Python machine learning.* Packt publishing ltd.

Rat, C., Hild, S., Sérandour, J. R., Gaultier, A., Quereux, G., Dreno, B., Nguyen, J.-M., et al. (2018). Use of smartphones for early detection of melanoma: systematic review. *Journal of medical Internet research*, 20(4):e9392.

Rigel, D. S., Friedman, R. J., Kopf, A. W., and Polsky, D. (2005). Abcde—an evolving concept in the early detection of melanoma. *Archives of dermatology*, 141(8):1032–1034.

Rossi, J. G., Rojas-Perilla, N., Krois, J., and Schwendicke, F. (2022). Cost-effectiveness of artificial intelligence as a decision-support system applied to the detection and grading of melanoma, dental caries, and diabetic retinopathy. *JAMA network open*, 5(3):e220269–e220269.

Simmons, C. and Holliday, M. A. (2019). A comparison of two popular machine learning frameworks. *Journal of Computing Sciences in Colleges*, 35(4):20–25.

Skvara, H., Teban, L., Fiebiger, M., Binder, M., and Kittler, H. (2005). Limitations of dermoscopy in the recognition of melanoma. *Archives of dermatology*, 141(2):155–160.

Sommerville, I. (2011). *Software Engineering, 9/E.* Pearson Education India.

Swanson, N. A., Lee, K. K., Gorman, A., and Lee, H. N. (2002). Biopsy techniques: diagnosis of melanoma. *Dermatologic clinics*, 20(4):677–680.

Udrea, A., Mitra, G., Costea, D., Noels, E., Wakkee, M., Siegel, D., de Carvalho, T., and Nijsten, T. (2020). Accuracy of a smartphone application for triage of skin lesions based on machine learning algorithms. *Journal of the European Academy of Dermatology and Venereology*, 34(3):648–655.

Wen, D., Khan, S. M., Xu, A. J., Ibrahim, H., Smith, L., Caballero, J., Zepeda, L., de Blas Perez, C., Denniston, A. K., Liu, X., et al. (2021). Characteristics of publicly available skin cancer image datasets: a systematic review. *The Lancet Digital Health*.

Yoon, C., Hamarneh, G., and Garbi, R. (2019). Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 365–373. Springer.

Zenone, M., Zocchi, L., Moccia, C., Passerini, S., Sanavia, T., Fariselli, P., Broganelli, P., Ribero, S., Maule, M., and Quaglino, P. (2022). Digital dermoscopy monitoring of melanocytic lesions: Two novel calculators combining static and dynamic features to identify melanoma. *Journal of the European Academy of Dermatology and Venereology*, 36(3):391–402.

Appendices

Appendix I: Confusion Matrices for Trained Models

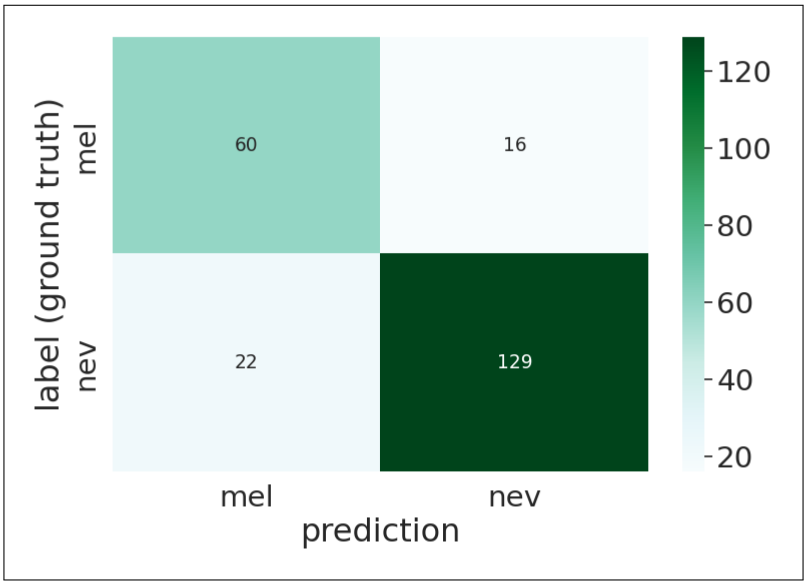


Figure 6.1: Confusion matrix for Resnet34

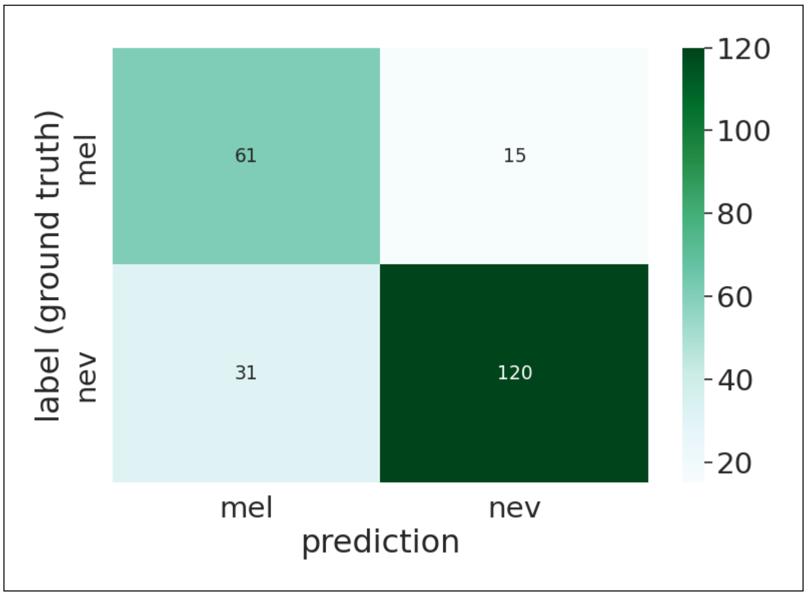


Figure 6.2: Confusion matrix for Resnet50

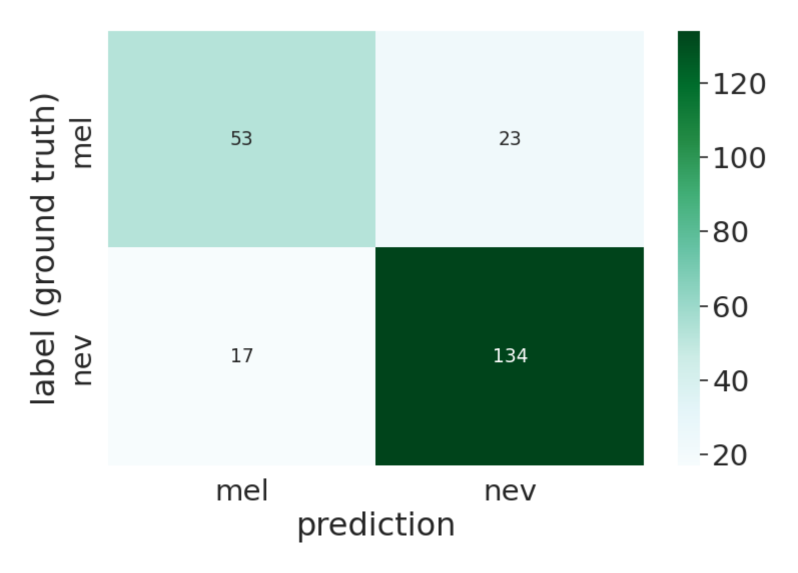


Figure 6.3: Confusion matrix for Resnet152

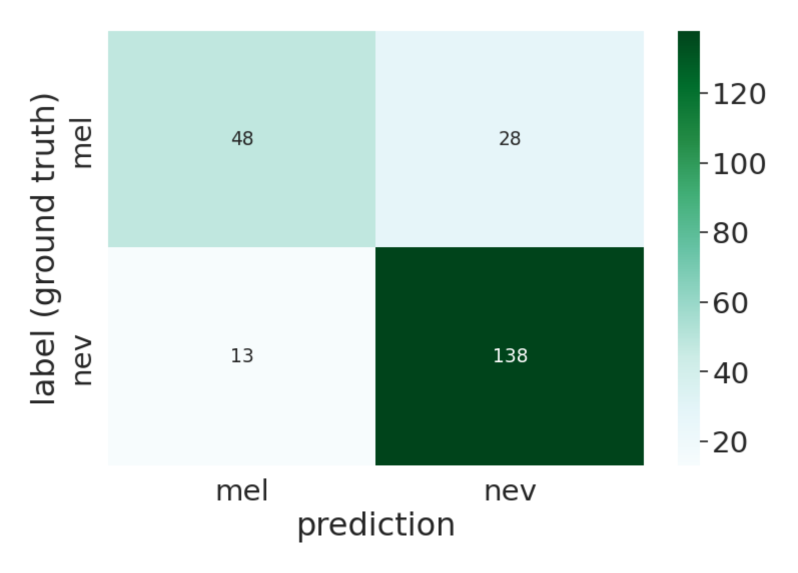


Figure 6.4: Confusion matrix for InceptionV3

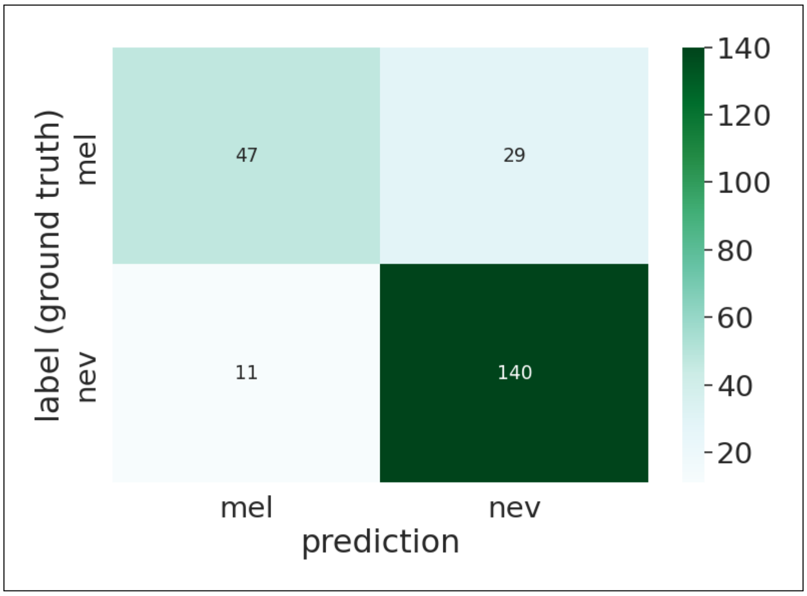


Figure 6.5: Confusion matrix for EfficientNetV2

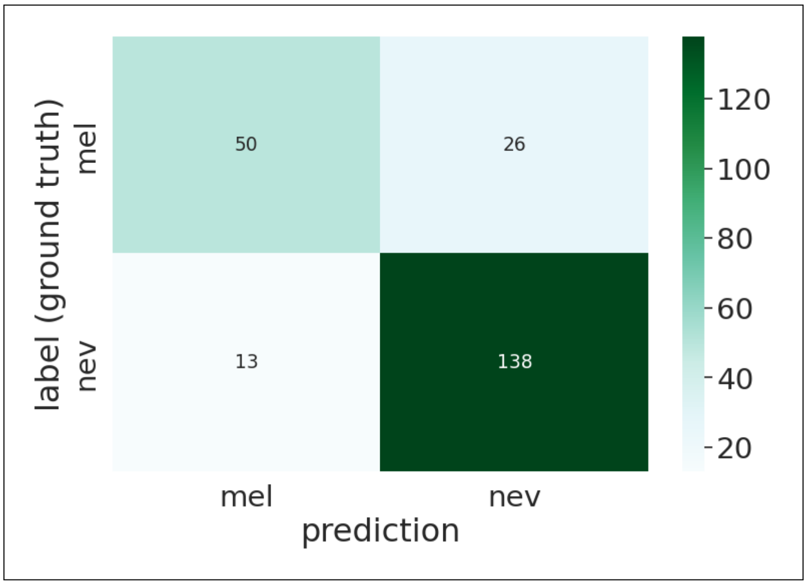
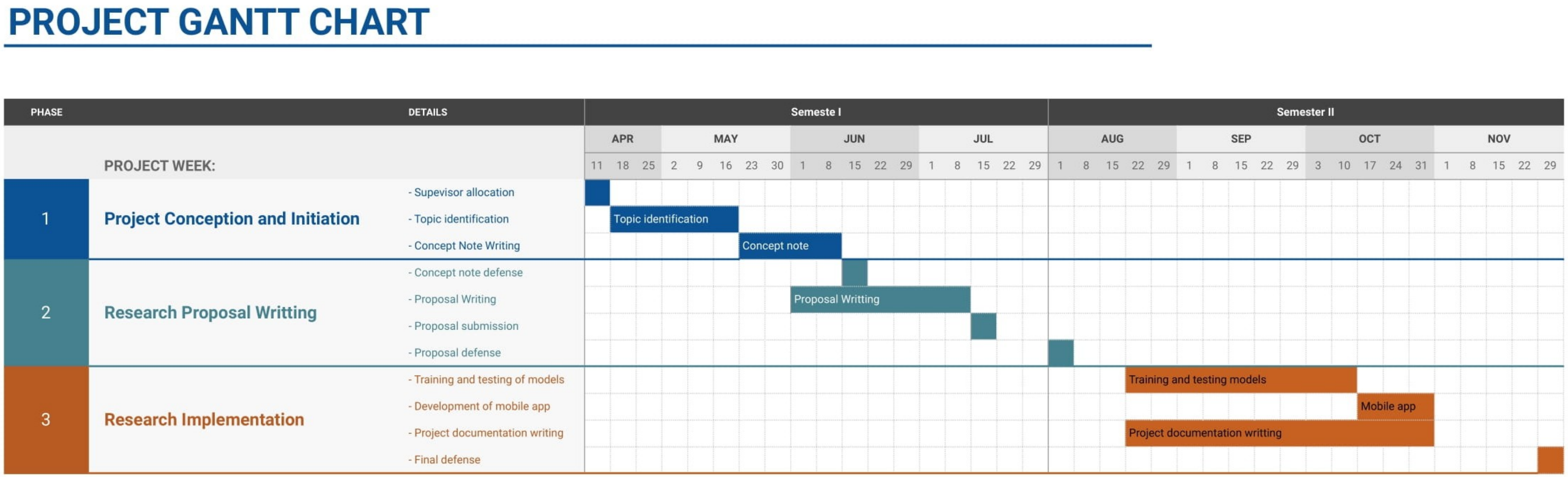


Figure 6.6: Confusion matrix for VisionTransformer

Appendix II: Gantt Chart



Appendix I: Gantt chart showing project timeline