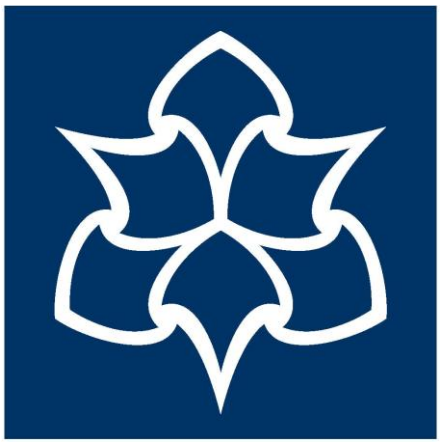
Comparative Analysis of Traditional Machine Learning and Deep Learning Approaches for Skin Lesion Classification

A DISSERTATION SUBMITTED TO MANCHESTER METROPOLITAN UNIVERSITY

FOR THE DEGREE OF BACHELOR OF SCIENCE

IN THE FACULTY OF SCIENCE AND ENGINEERING



May 2025

By

Bo Kwok

Department of Computing and Mathematics

Declaration

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project containing citations to the work of others, this project is my own unaided work. This work has been carried out in accordance with the Manchester Metropolitan University research ethics procedures, and has received ethical approval number 77470.

Signed: Bo Kwok

Date: 9th May 2025

Acknowledgements

I would like to express my sincere gratitude to my professors and tutors in Human-centred Computing Team, especially Dr Adrian K. Davison and Dr John Darby for their invaluable guidance, encouragement, and support throughout this research journey. Their insights and mentorship greatly enriched my academic experience.

Abstract

Skin cancer is a significant global health concern as its increasing occurrence and death rate. Early detection of malignant skin lesion is crucial for treatment as it can significantly decline the death rate to 23%. The ISIC 2024 SLICE-3D dataset is found to be distinctive in that, in additional to images dataset, it systematically includes metadata of physical lesion parameters, such as lesion size in mm lesion colour and contrast and irregularity, and 3D location data, which is directly derived from 3D Total Body Photography (3D-TBP). Therefore, this study evaluates and compares the performance of traditional machine learning and deep learning models for the classification of skin lesions using a relatively balanced subset of the ISIC 2024 SLICE-3D dataset. The methodology of this project is to preprocess the dataset to the ratio of 1.5 benign samples to 1 malignant sample, as malignant samples are too few (381 samples). The 1.5:1 ratio avoids the dataset being too small. This project implements six traditional machine learning models (Naïve Bayes, Decision Tree, Random Forest, XGBoost, LGBM, CatBoost) and four deep learning models (AlexNet, ResNet50, VGG11, ViT) which are commonly used in ISIC datasets. The models are trained and tuned in Google Colab, and evaluated using evaluation metrics such as test accuracy, F1 score, precision, recall, and ROC-AUC score. The results demonstrate that traditional machine learning models, especially gradient boosting algorithms (test accuracy: 84.4%-88.5% and F1 score: 84.4%-88.1%), outperform deep learning models (test accuracy: 56.2%-62.5% and F1 score: 53.8%-58.7%) across all evaluation metrics. Throughout all the models, XGBoost performed the best, it achieved 88.5% test accuracy, 88.1% F1 score. On the other hand, deep learning models underperformed than all traditional machine learning models. It may be due to limited malignant samples and computational resources constraints. Among the deep learning models used, ResNet50 (test accuracy: 62.5% and F1 score: 57.1%) performed the best in deep learning models but its performance is significantly lower than traditional machine learning models. The future work should focus on utilizing leveraging gradient boosting algorithms for analyzing their feature importance, exploring the possibilities to combine deep learning models as feature extractors and traditional machine learning models as classifiers. Trying out other deep learning architectures are possible options too. In addition, improving computational resources

and enlarging malignant samples are essential to improve deep learning models' performance for skin lesion classification.

Contents

List of Figures

List of Tables

Abbreviations

AlexNet→Alex Convolutional Neural Network

ResNet → Residual Neural Network

VGG → Visual Geometry Group Network

ViT → Vision Transformer

XGB → Extreme Gradient Boosting

LGBM → Light Gradient Boosting Machine

CatBoost → Categorical Boosting

Chapter 1 - Introduction

1.1 Introduction to Field Study

According to the World Health Organization (2024), cancer is the leading cause of death all over the world. Skin cancer is caused by different various factors such as UV radiation, genetic factors, poor lifestyle habits, and smoking. Within these factors, UV radiation contributes the most. The depletion of ozone layer lets more UV rays contacts the Earth's surface, which is harmful to human's skin and worsening the situation. It leads to the expectation that the number of skin cancer cases will rise significantly.

Early detection of skin cancer is crucial for patients' surviving rates. If melanoma is identified early, the survival rate of the patient will be as high as 99%. However, if the diagnosis of melanoma is delayed, the survival rate will decline to 23% (Skin Cancer Foundation, 2024). This displays the importance of early detection and prevention as both death rates and treatment cost increase rapidly when the skin cancer progresses from stage 1 to stage 4. When melanoma is detected early, skin cancer can be treated effectively with a minor surgery to remove the skin lesion, so that it can prevent the cancer spread around.

3D Total Body Photography (3D-TBP) is an advanced medical imaging technique that used to capture 3D and high resolution images of the entire surface of skin. The images used in ISIC 2024 SLICE-3D dataset is captured by this technique. 3D-TBP captures simultaneous photos from many angles by placing multiple cameras around the patient. After that, these simultaneous photos are merged to create a 3D model of the patient's body. This technology ensures that all visible lesions can be captured as photos for further diagnosis use (Rayner et al., 2018).

Both traditional machine learning and deep learning contribute a lot to medicine. Traditional machine learning can predict disease from analyzing the data of patients such as vital signs and blood test, they could predict diseases such as heart disease and diabetes

etc. It also detects patterns of patient's data to detect early cancer. It also supports clinical decisions by analyzing historical data and to suggest medicine to patients based on their profiles (Bozyel et al., 2024). Deep learning helps predicting functions of genes and associations of diseases from DNA sequences (Avsec et al., 2021), it also helps detecting diseases or cancers from images. It assists radiology by analyzing CT scans, X-rays and MRI results to detect abnormalities (Kumar, Sharma, & Gupta, 2024).

1.2 Aims and objectives

To develop a new technique for skin cancer detection, the aim of this project is to compare and evaluate both traditional Machine Learning and Deep Learning models for skin lesions classification using an ISIC dataset that includes metadata of physical lesion parameters, in additional to image dataset.

The objectives of this project are to:

1. find an ISIC dataset that includes metadata of physical lesion parameters, in additional to image dataset.
2. preprocess and balance the dataset by selecting similar amount of benign and malignant images to fulfil more balanced class distribution, while preventing the dataset being too small.
3. select suitable traditional machine learning models to implement the dataset
4. select suitable deep learning models to implement the dataset
5. evaluate and compare results of both traditional machine learning and deep learning models using evaluation metrics
6. Analyze the models' performance and providing insights about which model or approach is more suitable for skin lesion classification for further study.

1.3 Overview of report

The paper is separated into several chapters. The second chapter reviews
and mentions related work and literature reviews from other researchers, it includes the models' origin, and their usage on ISIC datasets or other medical datasets and the

hyperparameters used of model in their work. The third chapter displays the models' design, it starts with introducing the dataset which this projected utilized, explaining reasons of selecting this dataset. It displays the attributes, outliers and noise of the dataset. It introduces the 10 traditional machine learning and deep learning models which used to analyze the dataset. The fourth chapter explains the implementation of dataset and models, it shows how the dataset is prepared and pre-processed. It explains how the models apply to the dataset and the tuning of hyperparameters. At last, it listed the evaluation metrics and explains how these metrics evaluate the performance of the models. The fifth chapter presents the results and performances of models and discuss and analysis the experimental results and limitations. Finally, the sixth chapter concludes the paper and suggests recommendations for future research.

Chapter 2 - Literature Review

2.1 Traditional Machine and Deep Learning Algorithms used in ISIC or other medical datasets in previous research.

2.1.1 Comparison of Machine Learning Algorithms Used for Skin Cancer Diagnosis

The research used image data from the ISIC (International Skin Imaging Collaboration) database. This dataset contains skin lesions images that categorized to benign and malignant. 2,637 images were used for training and 660 images for testing. The images were resized to 224×244 pixels to ensure consistent input for analysis.

The methodology of this research was split into three stages, which are preprocessing and feature extraction, feature combining, and classification. Researchers convert images to grayscale or HSV (hue, saturation and value) and analyzes their shape, colour (with histograms) and texture (with Haralick and LBP descriptors). All these features are grouped as combinations to find the best diagnostic features in classification task.

All combinations of features were tested by each machine learning models to find the optimal feature set. Deep learning models were trained by Adam optimizer with 0.0005 learning rate for 20 epochs. Regularization and data augmentation are implemented to deep learning models to avoid overfitting.

Traditional machine learning models such as Random Forest, SVM, and Logistic Regression were compared with deep learning models with VGG-16, InceptionV3, Inception-ResNetV2, and ResNet-50. Traditional machine learning models performed better than deep learning models, especially with smaller datasets. Random Forest performed the best throughout the models, achieving 86.36% accuracy and 95% recall, using a combination of Histogram, LBP, and Haralick descriptors. Deep learning models especially Inception-ResNetV2 performed slightly worse than traditional machine learning. Naïve Bayes achieved 100% recall, but its accuracy was low.

The study concluded traditional machine learning models with proper feature selection could perform better than deep learning models in small datasets, while Random Forest is the most efficient and accurate model. The researchers suggest the preprocessing techniques can be added and more features can be expanded in the future research. The major limitation that the study suggested is the size of the dataset is too small. It causes overfitting for deep learning models. There is no segmentation implemented in the deep learning models, so the background maybe unclear and resulting outcomes were affected (Bistroń and Piotrowski, 2022).

2.1.2 A comprehensive analysis of deep learning and transfer learning techniques for skin cancer classification

ISIC 2018 dataset is utilized in this study, it contains 3,300 dermoscopic images classified as benign or malignant. The dataset was split with 8:2 ratio for training and testing data. The images were resized and denoised. It uses Contrast Limited AHE technique to improve the image quality.

The methodology includes preprocessing, segmentation, feature extraction and classification, and evaluation. resizing, denoising, and improving contrast are used for preprocessing the dataset. Segmentation helps identifying lesion boundaries by using active contour model. For feature extraction and classification phase, it used pre-trained deep learning models (VGG19, ResNet18, MobileNetV2) as feature extractors, then combines features and fed to traditional machine learning classifiers (SVM, KNN, DT, Naïve Bayes).

Every model was trained with 0.0001 learning rate, 64 batch size and maximum 25 epochs. ResNet18 and MobileNetV2 use Stochastic Gradient Descent with Momentum (SGDM) for optimization algorithm, while VGG19 used Root Mean Square Propagation (RMSProp). Both L2 regularization and categorical cross-entropy loss function were implemented to deep learning models to prevent overfitting and enhance their performance.

Deep learning alone showed good performance, accuracy improved when combining with traditional machine learning classifiers using extracted features. The hybrid

approach of combining features from multiple pre-trained models and using traditional machine learning classifiers outperformed using both deep learning models or traditional machine learning models individually. The best accuracy (92.87%) was achieved using features from ResNet18 and MobileNetV2 combined and classified with SVM.

This study demonstrated that combing deep learning models for feature extraction and traditional machine learning models to perform classification can improve the accuracy of detecting skin cancer. Although the study performed good results, its limitations still existed such as image quality issues and imbalance of datasets. The researchers suggested that they can include better hyperparameter tuning and extend their work of classification into multiple classes or real-time data (Shakya, Patel & Joshi, 2025).

2.1.3 My competition summary - ISIC 2024

Nlztrk (2024) summarized the performance of researchers who performed well in ISIC 2024 "Skin Cancer Detection with 3D-TBP" competition. The mostly used deep learning models are EfficientNet, Swin Transformer, ConvNeXt, EVA02, ViT and ResNet. For traditional machine learning models, gradient boosting algorithms such as LightGBM, CatBoost and XGBoost.

The main strategies utilized by top teams are combining gradient boosting algorithms with deep learning models to classify skin lesions. They integrate image features derived by deep learning with patient metadata and using out-of-fold (OOF) predictions as inputs for gradient boosting decision tree models. They mitigate class imbalance issues by employing heavy data augmentation and stratified k-fold during training. phase.

They also encountered some challenges such as overfitting to cross-validation as they heavily relying on early stopping during training phase, which leads to poor results. Some of the splits are in a mess, as some images from the same patient appeared in both training and validation sets, leading to unrealistically high performance. Researchers introduce redundant features leads to declined performance of models as the features collinear and become more complex.

2.1.4 Comparative study and analysis on skin cancer detection using machine learning and deep learning algorithms

This study used multiple public datasets of dermoscopic skin images, such as the ISIC archives from 2016 to 2020, HAM10000, PH2, and Med Node. These datasets contain images of different skin lesion types such as melanoma, benign nevi, basal cell carcinoma, and others.

The study utilized traditional machine learning and deep learning models. Traditional machine learning models are Support Vector Machine, KNN, and Random Forest. Deep learning models are using convolutional neural networks (CNNs). It used noise removal, colour correction and segmentation for preprocessing. Some of the CNNs are built from scratch, the others are from MobileNetv2, DenseNet, and InceptionV3. The images were fed into these deep learning models to classify lesions. The dropout rates and learning rates are tuned to reduce overfitting.

The results of this study show that deep learning models outperformed traditional machine learning models. Random Forest scored 87.32% in accuracy, while deep learning models like MobileNetv2 and ensemble networks scored up to 97.58%. As traditional machine learning models required manual feature extraction, while deep learning models learned features automatically, so traditional machine learning models may underperform.

The study founded out deep learning is effective for skin cancer detection, which may support early diagnosis potentially. On the other hand, as the datasets are imbalanced, leading to challenges during classifying similar lesions and unclear dark skin tones. It may affect the accuracy in real-world scenarios (Nancy et al., 2023).

2.1.5 Skin Lesion Classification Using Hybrid Deep Neural Networks

This study utilized ISIC 2016 and 2017 datasets for training. They contain 2,037 dermoscopic images, 411 of them are malignant melanoma, 254 them are seborrheic keratosis (SK), and the remaining images are benign.

The methodology is a hybrid approach, the researchers used deep learning models AlexNet, VGG16 and ResNet-18 to extract image features. The images are resized and normalized to match the models' dimensions, they are rotated and flipped for the data augmentation. After extracting features from images, the extracted features were fed into support vector machine classifiers. Results from each SVM were averaged, and probabilities were calibrated using logistic regression for predictions.

The best results were combining features obtained from all three deep learning models and classify by SVM, it scored an average AUC of 90.69%, with 83.83% for melanoma and 97.55% for SK. The result demonstrated that hybrid approach outperforms single deep learning or traditional machine learning model. Although the results are impressive, there are some limitations needed to be considered. The images are resized to small dimensions, it may lead to lose details. Limited training data is also a major limitation of this study (Mahbod et al., 2019).

2.1.6 Towards Skin Cancer Classification Using Machine Learning and Deep Learning Algorithms: A Comparison

The study utilized the International Skin Imaging Collaboration (ISIC) 2018 dataset. It includes 10016 instances and 1000 images with dermoscopic attributes. The dataset splits the types of skin cancer into seven categories: actinic keratoses, basal cell carcinoma, benign keratosis, dermatofibroma, melanoma, pigmented nevi, and vascular lesions. It includes five features and two meta attributes. It splits 40% of the data used for training and 60% for testing.

4 traditional machine learning and 1 deep learning models are applied in the study, which are Decision Tree, Naïve Bayes, Logistic Regression and K-Nearest Neighbour for traditional machine learning models, while convolutional neural network is used in deep learning models. Orange, an open-source machine learning toolkit, was implemented into the models. Models aimed to classify different types of skin cancer from the dataset.

Their performance was evaluated by metrics such as accuracy, precision, recall, F1 score, and AUC.

Convolutional neural network achieved the best performance throughout the models with 89% accuracy. For traditional machine learning models, Logistic Regression (88%) and Naïve Bayes (87%) followed closely, while Decision Tree achieved 86% accuracy and KNN scored the lowest at 81%. Researchers concluded that CNN is the best model for classifying skin cancers in the ISIC dataset. They emphasized that deep learning models outperformed traditional machine learning models. They suggested that using ensemble deep learning models and improved datasets may obtain a better accuracy (Kiran et al., 2021).

2.1.7 Advancing Skin Cancer Prediction Using Ensemble Models

This study used ISIC 2018 as dataset, it contains dermatoscopic images including seven types of skin lesions. As there are class imbalance, data augmentation is applied to create 3200 images per class approximately, ensuring the dataset is balanced. It utilized ensemble method to combine five traditional machine learning models together. They are Random Forest, Gradient Boosting, AdaBoost, CatBoost, and Extra Trees. they were separately trained and used Max Voting technique to vote their outputs to form the final prediction.

The Max Voting ensemble method scored an impressive accuracy of 96.20%, with precision of 96.30%, recall of 95.50%, and an F1 score of 95.63%. The study concluded that Max Voting ensemble approach improves skin cancer classification accuracy and reliability significantly, it may assist diagnosis potentially in the future (Natha and RajaRajeswari, 2024).

2.1.8 Vision Transformer for Skin Cancer Identification Based on Contrastive Learning and Adaptive-Scale Fragmentation

The study utilized ISIC 2019 dataset from Kaggle. It contains 10,015 dermoscopic images. These images are classified as seven types of skin cancer lesions: dermatofibroma, melanoma, actinic keratosis, basal cell carcinoma, nevus, vascular lesion, and pigmented benign keratosis. The images were divided into 70% training, 20% validation, and 10% testing sets. The images were reshaped and rescaled.

The researchers implemented an improved version of ViT model. They also introduced adaptive-scale image fragmentation to the model to better capture image features by allowing overlapping patches of multiple scales. The images were embedded into small patches and processed by ViT. The improved version of ViT was optimized with Adam optimizer with 0.0005 learning rate. Early stopping was implemented to the training phase of model to prevent overfitting.

Deep learning models Inception V3, MobileNet, and ResNet-50 are compared with improved version of ViT's performance. Inception V3, MobileNet, and ResNet-50 scored 72%, 94.3% and 89% accuracy respectively, while improved version of ViT model achieved 99.66% accuracy, 94.85% precision, 93.74% recall, and 94.52% F1 score. It demonstrated that transformers models can capture complex features better than other deep learning models in medical image datasets.

The researchers concluded that the improvement of ViT model may enhance automated skin cancer diagnosis and provides more accurate results than other methods (Naeem et al., 2024).

2.1.9 Skin Lesion Classification Using a Deep Ensemble Model

This study utilized ISIC 2018 and HAM10000 dataset, ISIC 2018 contains 2357 dermoscopic images categorized into different skin cancers. The study only extracted melanoma, basal cell carcinoma, and squamous cell carcinoma from ISIC 2018 and HAM10000 dataset as subset to analyze. The oversampling method was applied to handle class imbalance.

The researchers combined VGG16, Inception-V3, and ResNet-50 to form a skin lesion classification system. The images are normalized and resized to 224×224 or 299×299

pixels, which depends on the input demand of different models. The ensemble model merged three models' predictions using a weighted average.

The ensemble model scored 97% accuracy in the subset, which performs better than VGG16, Inception-V3, ResNet-50 models individually. The results proved that the ensemble model is effective to classify skin lesions. Using oversampling can mitigate the effect of class imbalance. Researchers suggested that the model can assist detecting skin cancer (Thwin and Park, 2024).

Based on the review of nine literature studies, six traditional machine learning models (Naïve Bayes, Decision Tree, Random Forest, XGBoost, LGBM, CatBoost) and four deep learning models (AlexNet, ResNet50, VGG11, ViT) are commonly used in ISIC datasets and are employed in this research.

2.2 Origin of Traditional Machine and Deep Learning Algorithms models used

2.2.1 Naïve Bayes

Naïve Bayes model is from the concept of Bayes' Theorem in 1763 with its assumption of feature independence. Maron (1961) developed the Naïve Bayes with wide-mentioned model which applies Bayes' Theorem. Naive Bayes assumes conditional independence between features, which is expressed in:

$$P(X \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

So, the classification using Naïve Bayes model would be:

$$\hat{C} = \arg\max_{C} P(C) \prod_{i=1}^{n} P(x_i \mid C)$$

2.2.2 Decision Tree

The concept of decision trees is invented in 1963 by Morgan and Sonquist (1963), it is named as Automatic interaction detection (AID) and mainly used in statistics. AID creates tree structures to maximize explained variance at each split. This method mainly in analyzing the difference between splits and required large datasets to analyze, so it was not popular and widely used in nowadays. ID3 is introduced by Quinlan (1986), ID3 develops simple and efficient trees, it can learn from small samples so it does not require large datasets to apply. ID3 applies heuristic approach that it takes the attribute which provides the highest information gain at each step, this theory is still used in decision tree nowadays (Quinlan, 1986).

2.2.3 Random Forest

Random Forest algorithm is invented by Breiman (2001), Breiman's Random Forest original algorithm combines a collection of tree-structured class predictors, each tree is assigned to a value of random vector sampled independently, while the tree should be evenly distributed in the forest, then each tree votes for the most popular class. The Random Forest algorithm contains the idea of bagging (Breiman, 1996) and the random selection of features, it is included in the random subspace method which introduced by Ho (1995). Both methods assisted to construct bunches of tree-structured predictors with controlled variance (Breiman, 2001).

2.2.4 XGBoost

XGBoost is invented by the Distributed (Deep) Machine Learning Community (DMLC) in University of Washington, the aim of inventing XGBoost is to push the limit of computational power by constructing an algorithm which is scalable, portable and

accurate. XGBoost is widely adopted in machine learning and wins many machine learning competitions (Chen & Guestrin, 2016).

XGBoost original model is a machine learning system operated by boosting trees. It implements Gradient Boosting framework into its algorithm. It delivers a parallel tree boosting which processes data science tasks. It supports major programming languages such as Python, C++ and R and applies in some distributed computation environments such as Hadoop and MPI (Chen & Guestrin, 2016).

Due to the popularity of GPU in nowadays, XGBoost improved its speed when using GPU and multi-threads. It also supports more programming languages such as Scala and Julia.

2.2.5 LGBM

LGBM, as known as LightGBM, is developed by one of the Microsoft Research team in Asia in 2017.

LGBM original model is a tree-structured algorithms which implements novel gradient boosting framework, it is good at processing large data. It implements two novel techniques, which are Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), to reduce number of data instances and features respectively. GOSS does random samplings on small-gradient data instances, while it saves the instances with large gradients, this approach helps to retain the accuracy of information gain. EFB reduces the number of effective features by bundling mutually exclusive features. These two techniques help the original model of LGBM train faster and more efficiency with keeping the similar accuracy (Ke et al., 2017).

LGBM's latest version does not have any significant difference, benefiting from the popularity of GPU and CUDA, the training process are accelerated.

2.2.6 CatBoost

CatBoost is developed by Prokhorenkova et al. (2018), aims to solve the problem of other gradient boosting methods about target leakage of prediction shift when handling categorical variables. This target leakage happens then the model accidentally used the target variable when training, it causes bias on prediction. CatBoost implements techniques to solve this issue to let the model more reliable and unbiased.

CatBoost original model innovates ordered boosting, it improves the gradient boosting algorithm from other models by implementing a permutation driven approach. It processes the data in a specific order to prevent the target leakage by avoiding the model using the future information. Catboost calculates statistics related to categorical features in an ordered manner, instead of using the whole dataset to calculate. This approach can make sure that the data point's computation does not include its own target value, it can prevent target leakage (Prokhorenkova et al., 2018).

CatBoost nowadays supports more types for loss functions, mainly related to GPU. It supports more platforms such as spark and Cuda-related platforms.

2.2.7 AlexNet

AlexNet was developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton at the University of Toronto in 2012. The researchers wanted to develop a large and deep convolutional neural network which can trained on a GPU hardware and can analyze on huge image datasets such as ImageNet in a high accuracy. The researchers also wanted to reduce the error rate in a large instance (Krizhevsky, Sutskever and Hinton, 2012).

The architectures of AlexNet original model contains regularization technique, activation function, optimizers, pooling etc. These architectures will be discussed later.

AlexNet nowadays improved their memory and computational power, thanks to the advancement of GPU. Batch normalization has been implemented into AlexNet for better regularization, while Advanced optimizers such as Adam optimizer are used as an improved optimization algorithm.

2.2.8 ResNet

ResNet is developed by He at el. (2016), the developers of ResNet wanted to mitigate the difficulties of training process for deeper neural networks, as when deep neural networks are trained, their performance are usually worse in training and testing error.

To solve this problem, the developers implemented the residual learning framework into the neural network and became the original model of ResNet. Residual learning framework lets each few stack layers fit in a residual mapping, instead of fitting them in a desired underlying mapping as other deep neural networks. They found out it is easier to optimize the residual mapping, than the original mapping. The architecture of ResNet include 5 different models, including ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152, the numbers represent the layers of ResNet, while ResNet-34 and ResNet-50 are the most notable models as they display the key architectural innovations in the original paper of ResNet (He at el., 2016).

Due to the popularity of GPU nowadays, ResNet improves its memory and computational power significantly, while dropout and label smoothing enhances its regularization, Adam optimizer also increases ResNet's optimization.

2.2.9 VGG

VGG is developed by Karen Simonyan and Andrew Zisserman from the Visual Geometry Group at the University of Oxford in 2014. The developers of VGG investigates the effect of the depth of convolutional neural networks on their accuracy in large image recognitions tasks. They suggested that deeper networks with very small convolutional layers such as 3x3 may improve results. In this background, the original model of VGG was produced.

The original model of VGG contains 4 models, which are VGG-11, VGG-13, VGG-16, VGG-19, the numbers represent the layers of VGG. The basic architectures are the

convolutional and max pooling layers must be 3x3 and 2x2 respectively, fully connected layers and softmax activation functions is put at the end.

In nowadays, some architecture of VGG is different from its original model. It replaces some fully connected layers with global average pooling, it also implements batch normalization layers to improve training stability (Simonyan and Zisserman, 2015).

2.2.10 ViT

Vision Transformer (ViT) is developed by Dosovitskiy et al., (2021), a research team of Google Research. The developers wanted to test whether the inductive biases in convolutional neural networks are important, and whether a model relies only on attention mechanisms can perform better than CNNs. ViT is produced to test these hypotheses.

The characteristics of the original model of ViT is splitting the image into 16x16 fixed-size patches. It flattens every patch and puts it into embedding space. After that, it processes the patches similarly to NLP sequences.

As ViT is a new Deep Learning architecture, there aren't any significant differences between the original model and the latest model, the significant improvement is pooling strategies, global average pooling and multi-head attention pooling is used in ViT to improve its performance.

Chapter 3 - Design

3.1 Justification of Dataset Selection

ISIC 2024 - Skin Cancer Detection with 3D-TBP SLICE-3D dataset is utilized in this project, it is a medical dataset of skin lesions developed for ISIC 2024 challenge. This 400000 image dataset aims on detecting skin cancer. It contains images that cropped to 15mm x 15mm in the field of view, while the skin lesions have been centred, and they are extracted from the technology of 3D Total Body Photography (3D-TBP). The images have been provided by hospitals and universities all over the world, and the dataset were curated by International Skin Imaging Collaboration (International Skin Imaging Collaboration, 2024).

The main reason that this dataset is selected because ISIC 2024 dataset is the only large medical dataset provides metadata of physical lesion parameters completely, such as lesion size in mm lesion colour and contrast and irregularity, and 3D location data. Using this dataset can compare the performance of deep learning models by processing skin lesion image and the performance of traditional machine learning models by processing the numerical data to find out whether traditional machine learning or deep learning models, and a particular model have a better performance on analyzing ISIC 2024 dataset.

ISIC datasets is a very popular and trustable dataset which is widely used by dermatology AI research, it involves the early detection of skin cancer, which sticks to real-world clinical scenarios.

3.2 Attributes of dataset

The attributes of the ISIC 2024 dataset are isic_id, target, patient_id, age_approx, sex, anatom_site_general, clin_size_long_diam_mm, image_type, tbp_tile_type, tbp_lv_A, tbp_lv_Aext, tbp_lv_B, tbp_lv_Bext, tbp_lv_C, tbp_lv_Cext, tbp_lv_H, tbp_lv_Hext,

tbp_lv_L, tbp_lv_Lext, tbp_lv_areaMM2, tbp_lv_area_perim_ratio, tbp_lv_color_std_mean, tbp_lv_deltaA, tbp_lv_deltaB, tbp_lv_deltaL, tbp_lv_deltaLB, tbp_lv_deltaLBnorm, tbp_lv_eccentricity, tbp_lv_location, tbp_lv_location_simple, tbp_lv_minorAxisMM, tbp_lv_nevi_confidence, tbp_lv_norm_border, tbp_lv_norm_color, tbp_lv_perimeterMM, tbp_lv_radial_color_std_max, tbp_lv_stdL, tbp_lv_stdLExt, tbp_lv_symm_2axis, tbp_lv_symm_2axis_angle, tbp_lv_x, tbp_lv_y, tbp_lv_z, attribution, copyright_license, lesion_id, iddx_full, iddx_1, iddx_2, iddx_3, iddx_4, iddx_5, mel_mitotic_index, mel_thick_mm and tbp_lv_dnn_lesion_confidence.

The attribute target is used to determine whether the lesions are benign (0) or malignant (1). The unique identifiers for patients, lesions and cases are patient_id, lesion_id and isic_id respectively. Clinical and demographic attributes are age_approx, sex, tbp_lv_location and tbp_lv_location_simple, age_approx and represent approximate patient age and sex respectively, tbp_lv_location and tbp_lv_location_simple represent classification of anatomical location. The attributes of lesion characteristics are clin_size_long_diam_mm, tbp_lv_minorAxisMM, tbp_lv_areaMM2, tbp_lv_perimeterMM, and tbp_lv_area_perim_ratio, which represents maximum diameter, smallest diameter, area, perimeter border jaggedness of skin lesion respectively. Colour and contrast attributes are tbp_lv_A, tbp_lv_B, tbp_lv_C which are inside lesions, and their external counterparts (tbp_lv_Aext, tbp_lv_Bext, tbp_lv_Cext), while tbp_lv_H and tbp_lv_Hext representing hue inside and outside the lesion respectively. Its values range from 25 for red to 75 for brown. The contrast metrics such as tbp_lv_deltaA, tbp_lv_deltaB, tbp_lv_deltaL, and tbp_lv_deltaLBnorm which ranges from 5.5 to 25 measure the contrast between lesion and its immediate surrounding skin. Irregularity is measured with tbp_lv_color_std_mean for colour variance, tbp_lv_norm_color for colour variation, tbp_lv_radial_color_std_max for colour asymmetry, and tbp_lv_norm_border for border irregularity. The structural attributes are tbp_lv_eccentricity, tbp_lv_symm_2axis, which is the asymmetry of border, and tbp_lv_symm_2axis_angle which is asymmetry angle of lesion border. tbp_lv_stdL and tbp_lv_stdLExt represents the standard deviations of lightness inside and outside lesion respectively. tbp_lv_nevi_confidence and tbp_lv_dnn_lesion_confidence represent confidence score of nevus and lesion respectively, while tbp_lv_x, tbp_lv_y, tbp_lv_z represents 3D coordinates. Diagnoses are recorded in iddx_full, further specific

diagnoses which presented in levels are recorded iddx_1 to iddx_5. Metric involves melanoma are mel_mitotic_index and mel_thick_mm, which mel_thick_mm represents the thickness of melanoma invasion. tbp_tile_type, attribution represents lighting modality and image source respectively, which joins with image_type, copyright_license are categorized with metadata.

3.3 Outliers and noise

Outliers usually appear in real-world datasets, so handling them in proper is very important before using machine learning models to train the datasets. ISIC 2024 - Skin Cancer Detection with 3D-TBP SLICE-3D dataset contains 401,059 rows and 55 columns, with a mix of numeric and text features. Interquartile Range is utilized in detecting outliers in numeric fields, it detected a certain number of outliers in lesion measurements and model confidence score. For example, tbp_lv_dnn_lesion_confidence contains 80,379 outliers (20.0%), tbp_lv_areaMM2 has 36,748 outliers (9.2%), clin_size_long_diam_mm has 27,264 outliers (6.8%) etc. This shows that outliers are not ordinary cases and can potentially affect various analyses if they are not handled well.

To perform classification tasks, all 381 malignant samples and a random 571 benign samples are chosen to form the subset of the data om 952 samples. In this subset, tbp_lv_dnn_lesion_confidence had 200 outliers (21.0%), tbp_lv_areaMM2 had 114 outliers (12.0%) and clin_size_long_diam_mm had 75 outliers (7.9%). These statistics prove that outliers still remain in the subset and should be taken into consideration during model development.

Although not all traditional machine learning models are heavily affected by outliers, some of the models can be skewed significantly by the presence of outliers. However, the application of method depends on the aims of the project. In this study, the major aim is to compare the performance of deep learning models that applied to image data with traditional machine learning models that applied to metadata which extracted from those images. Both datasets should be treated the same measures as it ensures a fair comparison between deep learning and traditional machine learning approaches. If outliers are

removed or processed in only one dataset but not the other, the results may become biased. Therefore, for this case, outliers should be remained in both image and metadata datasets.

Keeping outliers in both datasets reflects real-world conditions during comparison, as outliers in the real-world often exist and can't be avoided. This approach ensures that the performance of deep learning and traditional machine learning models are not artificially influenced by pre-processing choices. These models should be trained by raw datasets in this study.

So, outliers were not removed in this project to create a fair and consistent environment for comparison between deep learning on image data and traditional machine learning on extracted metadata. Future work may explore the effect of outlier removal on model performance.

For noises, there are no incorrect labels or typos in the dataset. However, there are 19145 rows out of total 401059 rows containing missing values, while there are only 12 out of 393 malignant samples have missing values. Deleting missing values per list from the dataset is the most appropriate approach to minimize the effects of missing values when handling noise. For columns which have many missing values (>70%) such as 'iddx_2', 'iddx_3', 'iddx_4', 'iddx_5', and 'mel_mitotic_index', column-wise deletion is applied to these columns.

Image augmentation techniques mitigate the noise in the image dataset by cropping, resizing and normalizing pixel values to improve model performance.

3.4 Traditional Machine and Deep Learning Models used

A. Traditional machine learning models used in this project

1. Naïve Bayes (Gaussian)

Naïve Bayes classifier is based on Bayes Theorem in 1763, it assumes that all features are independent of each other. It first calculates the prior probabilities of each class and likelihood of feature values, then it calculates the posterior probability for each class and

predicts result by picking the class with highest posterior probability (Rish, 2001). Gaussian Naïve Bayes classifier is chosen for this project, as it presumes the features are normal distribution. This type of Naïve Bayes classifier performs better in processing non-binary datasets compared to other Naïve Bayes classifiers.

2. Decision Trees

Decision Trees algorithm is one of the traditional machine learning algorithms which is good at performing classification and regression tasks. In a decision tree, a node represents an attribute or feature, a branch represents a decision which based on feature values, branches usually would appear as larger or smaller, or yes or no. A leaf node represents the output. Decision Trees algorithm first runs through the whole data, finds the most important feature of the data using Gini Impurity, and use the most important feature to split the data. It repeats the process for each split subset until all data in a node belongs to the same class. If the data is too large, it will stop earlier when it reaches the max depth (Quinlan, 1986).

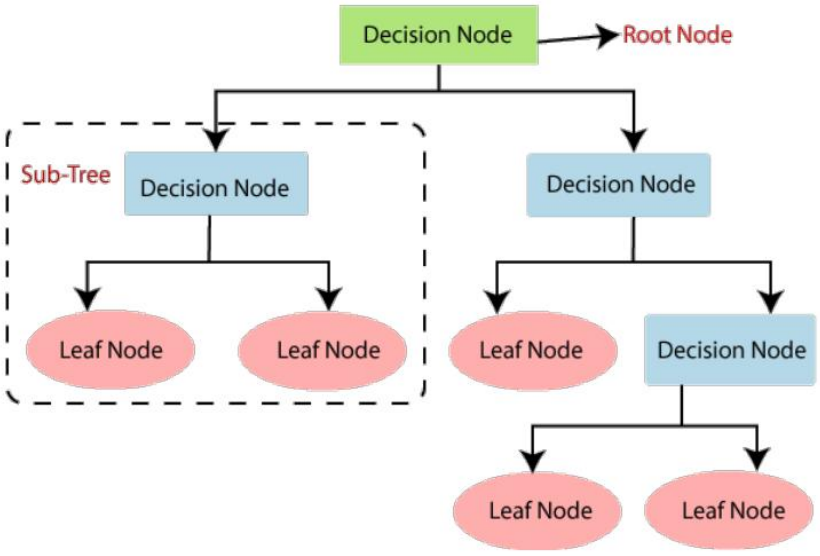


Figure 3.1: Decision Tree Model Architecture. Diagram re-used from (Hafeez et al., 2021)

3. Random Forest

Random Forest is a popular traditional machine learning algorithms that used to process classification and regression tasks. It combines the results of multiple decision tree to enhance the performance of its results. Random Forest can be classified as ensemble learning as it combines multiple models during its learning process. Random Forest first creating random subsets from the dataset, then assigns a subset to a decision tree for training. It implements feature randomness to avoid using same top features when learning the dataset, it achieves certain diversity for multiple decision trees. Each decision tree would have its own prediction, they vote their prediction to generate the final result of Random Forest algorithm (Breiman, 2001).

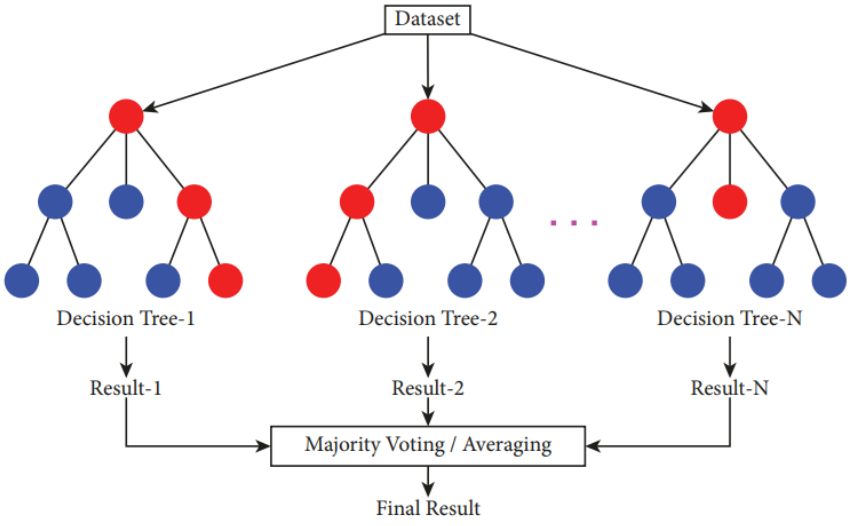


Figure 3.2: Random Forest Model Architecture. Diagram re-used from (Khan et al., 2021)

4. XGBoost

XGBoost, as known as Extreme Gradient Boosting, is a popular traditional machine learning algorithm, it is good at finding feature importance, processing classification and regression tasks. XGBoost implements decision trees with their gradient boosted, it builds the model one by one, as current model can correct the error from the previous

model. This approach can enhance its processing speed and performance. XGBoost start with a simple prediction to initialize the model, then every prediction is measured by a loss function, which is the difference between the prediction and actual values. Within the loss function, XGBoost takes gradient and hessian as the main gradients to calculate the loss in order to improve its performance. XGBoost utilizes gradients to fit a decision tree, it defines splits in the data that minimize gradients and hessians in the loss function. When the best tree is built, the predictions are scaled by a learning rate, learning rate can be set by users, it controls the influence of each tree. After that, it is added to the model. At last, XGBoost keeps repeating the process to build more trees until (Chen & Guestrin, 2016).

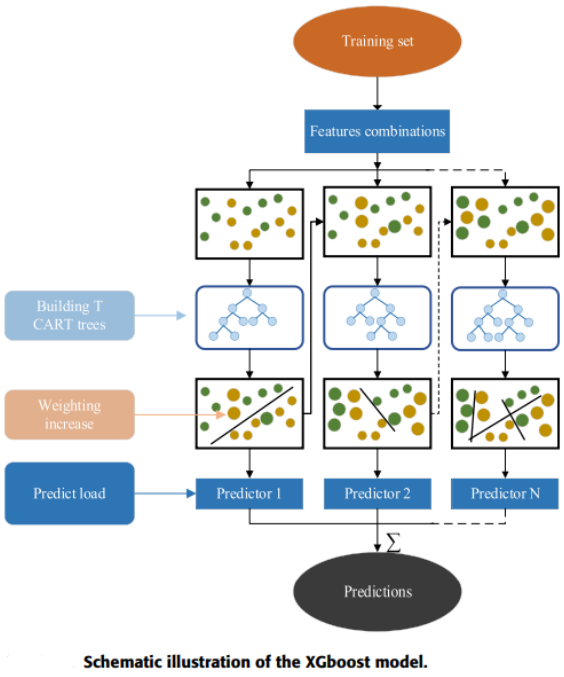


Figure 3.3: XGBoost Model Architecture. Diagram re-used from (Yao, Fu and Zong, 2022)

5. LGBM

LGBM, short for Light Gradient Boosting Machine, is based on decision tree algorithm which utilizes gradient boosting framework. It is utilized in varieties of traditional machine learning tasks. It handles large datasets well with speed and efficiency. Similar

to XGBoost algorithm, LGBM also implements gradient boosting, but it has a different approach. It first converts continuous features into discrete bins in a histogram, feature values will be assigned into the bins based on their range. This approach can reduce the usage of memory and enhance computation speed. Then it generates an initial prediction using log-odds for binary classification, it calculates the gradient and hessian in the loss function. Next LGBM finds the leaf with the largest loss and splits it, its approach can create trees with deeper depth, that allows LGBM itself can process data with complex patterns. After that LGBM updates its model with new tree and repeats until it reaches the maximum epochs or the criteria of early stopping (Ke et al., 2017).

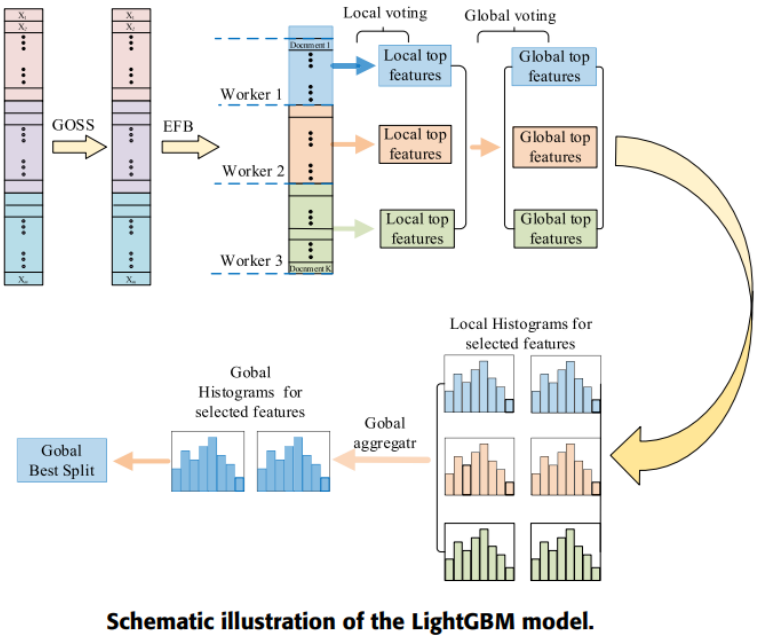


Figure 3.4: LightGBM Model Architecture. Diagram re-used from (Yao, Fu and Zong, 2022)

6. CatBoost

CatBoost, aka Categorical Boosting, is similar to XGBoost and LGBM as it uses gradient boosting framework to implement decision trees. It is good at handling categorical features and perform well in feature ranking, classification and regression tasks. It doesn't need any extensive preprocessing as it can process categorical data automatically. It utilizes ordered boosting algorithm to reduce overfitting. CatBoost

starts with encoding categorical features with ordered target encoding. It rearranges the data and calculate statistics on data points. CatBoost puts the data into a data pool, a data pool includes labels, features, categorical feature indices etc. It can optimize the memory usage and provide efficiency in data access. CatBoost first generates a basic prediction, if it is a classification task, the prediction would be log-odds. Then it loops to build decision trees using ordered boosting, it calculates the gradients and hessians by using the loss function and constructs a tree that aims to minimize the loss by selecting splits. After that, it updates its predictions with the new tree, which affects by the learning rate. Finally, CatBoost repeats the same procedure until it reaches the required number of loops or the criteria of early stopping (Prokhorenkova et al., 2018).

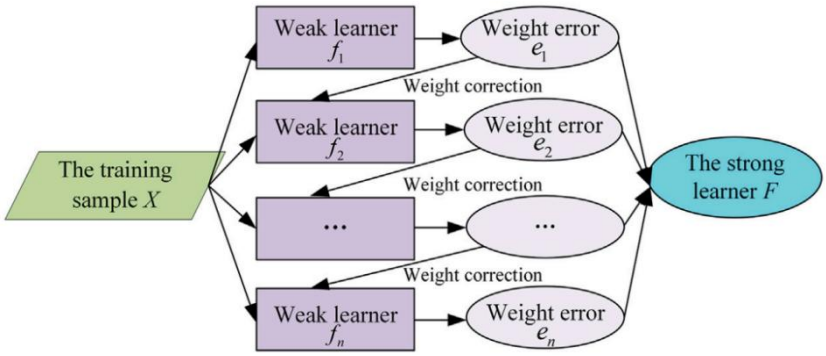


Diagram of CatBoost structure.

Figure 3.5: CatBoost Model Architecture. Diagram re-used from (Chen *et al.*, 2023)

B. Deep learning models used in this project

1. AlexNet

AlexNet is a Deep Convolutional Neural Network (DCNN) architecture which contributes the most to the deep learning revolution. It is a deep learning model which is designated for image classification. AlexNet's architecture contains eight layers, it has five convolutional layers and three fully connected layers. These layers can classify images over thousands of categories. AlexNet utilized ReLU, as known as Rectified Linear Unit activation function, it enhances the speed during training process. AlexNet's

max pooling technique helps to reduce the dimensions of the data, its dropout layers disable neurons randomly during training phase to reduce overfitting. It processes input images with RGB three colour channels. The convolutional and pooling layers of AlexNet allows it to extract numerous complex features from images. Features maps, which is the output from these layers, are flattened into one-dimensional vector, and passed through the fully connected layers. Softmax classifier is the final layer which calculates and displays probabilities for each possible class label (Krizhevsky, Sutskever and Hinton, 2012).

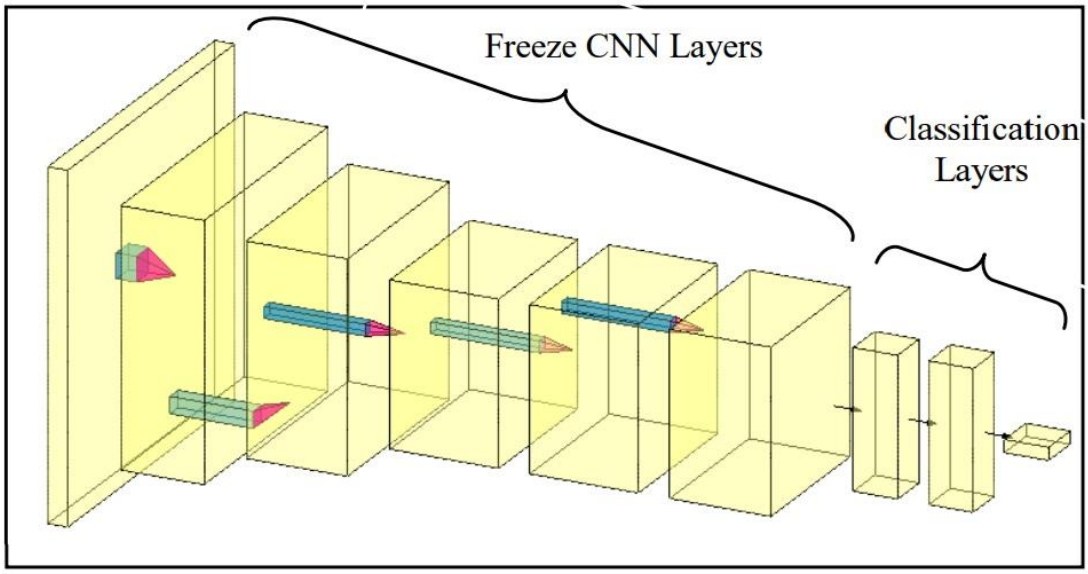


Figure 3.6: AlexNet Model Architecture. Diagram re-used from (Anilkumar, Velaga and Devi, 2019)

2. ResNet

ResNet, as known as Residual Network, is a deep learning convolutional neural network architecture. It is produced to train deep neural networks effectively. It can also perform image classification and computer vision tasks. It implements residual connections, which allows the network to skip layers using residual blocks to avoid optimization difficulties. When adding more layers into a deep network, the gradients may be extremely large or small, this situation affects the network updating the weights effectively, so the network may not learn effectively and result in performing worse. Residual networks can mitigate the vanishing gradients and performance issues. Its main

components are residual blocks, shortcut connections and architecture. Residual blocks are the building blocks of ResNet. It stores residual functions, which contains batch normalization, convolutional layers and ReLU function in it. Shortcut connections allow the network to skip layers and allows the input to add to the output of a residual block directly (He et al., 2016). There are different depths for the ResNet's architecture, ResNet-50 is chosen for this project. 50 represents the number of the total layers. Apart from ResNet-50, there are remaining 5 types of ResNet: ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, and ResNet-1202.

The 50 layers in ResNet-50 are split as 6 components. Its initial layer contains 7x7 convolutional layers, batch normalization, ReLU function and a 3x3 max pool technique. Stage 1 contains 2 residual blocks and outputs 256 channels. Stage 2 contains 4 residual blocks and outputs 512 channels. Stage 3 contains 6 residual blocks and outputs 1024 channels. Stage 4 contains 3 residual blocks and outputs 2048 channels. The final layer contains Global average pooling technique, Softmax classifier and fully connected layer, which can classify 1000 distinct categories. All residual blocks contain 3 layers, by adding the initial and final layer, they combine into 50 layers.

ResNet model starts with inference phase, the input is normalized and passes through the initial layer. Starting from stage 1, the residual blocks process the data and combines the outputs of convolutional layers with the input using shortcut connections. The global average pooling technique and a fully connected layer help produce class probabilities by using Softmax classifier. Next in the training phase, ResNet utilizes the loss function to compare whether the results are correct. Gradients are also calculated by backward propagation of errors, while weights are updated using optimizers, mean and variance are also updated by batch normalization in the training phase (He et al., 2016).

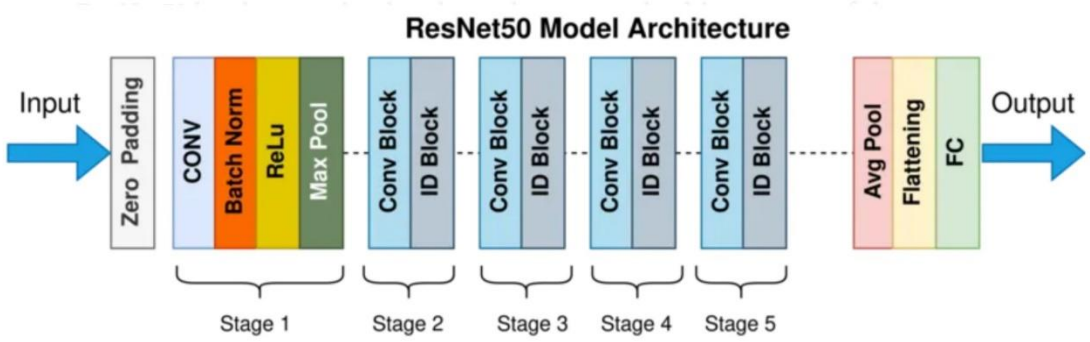


Figure 3.7: ResNet Model Architecture. Diagram re-used from (Kundu, 2023)

3. VGG

VGG, short for Visual Geometry Group, is one of the convolutional neural networks. It is good at performing image classification tasks. VGG uses 3x3 convolutional layers, 2x2 max-pooling layers and fully connected layers. VGG11 is used in this project, it contains 8 convolutional layers and 3 fully connected layers. VGG first applies 3x3 convolutional filters to process spatial dimensions. Each convolution filter contains a ReLU activation function to implement non-linearity. Multiple convolution layers are stacked before max pooling in order to increase the feature extraction depth. Next, VGG uses max-pooling to cut spatial dimensions to half, it allows the network to be more efficient in computational resources and reduce overfitting. After convolutions and pooling procedures, feature maps generated are flattened into a vector. It will go through the fully connected layers. After that, the input images pass through all layers, each layer transform the input into designated size and dimension of feature maps, and extract abstract features. In training phase, VGG uses backpropagation, which is backward propagation of errors in short, and stochastic gradient to minimize the loss function. At last, VGG generates a probability distribution of classes as output for classification tasks (Simonyan and Zisserman, 2015).

Figure 3.8: VGG Model Architecture. Diagram re-used from (Rath, 2021)

4. ViT

ViT is a deep learning model that applies the Transformer architecture unlike from other deep learning models that mentioned above. ViT treats images as sequential patches to process by using self-attention mechanisms. For image patching, the input image is divided into numerous smaller sizes of patches, the patches are flattened into a vector. Next, a fully connected layer transformers flattened patches into fixed size embedded vector and creates patch embeddings sequentially. Positional encodings are added into patch embeddings in order to keep the information about their image positions, as Transformers do not understand the spatial dimensions and positions of patches. Transformer encoder layers include Multi-Head Self-Attention, Feed-Forward Neural Network, Layer Normalization and Residual Connections. Multi-Head Self-Attention calculates attention scores to the relationships of model across all patches, it assists the model to capture global dependencies. Feed-Forward Neural Network applies a fully connected network to tokens. Layer Normalization and Residual Connections helps to stable the training process and enhance gradient flow. Transformer encoder layers extract and process the sequence of patch embeddings in parallel. Finally, the output of tokens is processed through a linear layer to generate class probabilities (Dosovitskiy et al., 2021).

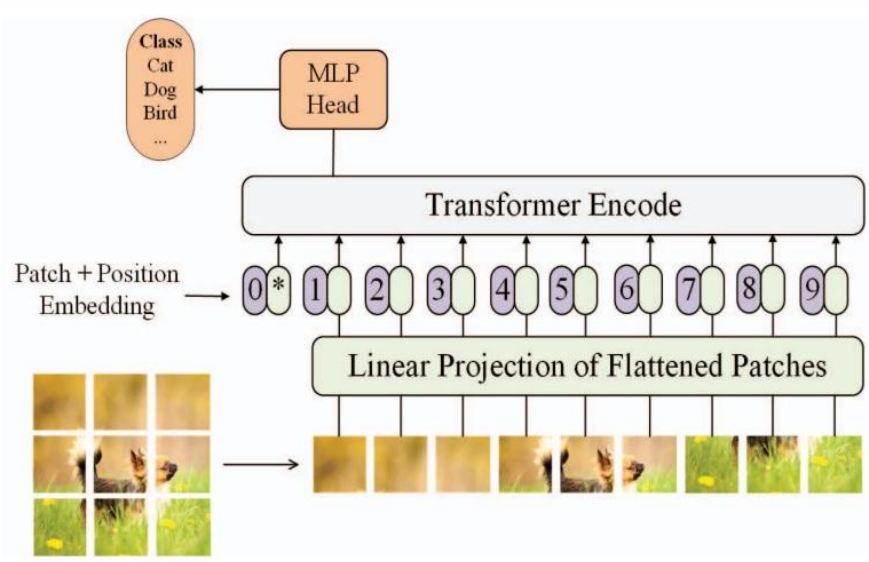


Figure 3.9: ViT Model Architecture. Diagram re-used from (Huo *et al.*, 2023)

Chapter 4 - Implementation

4.1 Data Preparation

This challenge is to detect skin cancer, which uses 3D Total Body Photography (3D-TBP) to crop the images. The dataset contains 400,000 15mm x 15mm cropped images, the distinct lesions have already been centralized in the images, so image centralization techniques such as bounding box detection are not used in this project. Within the huge number of images, there are only 393 of them are identified as malignant skin lesions, the rest of the images are identified as benign skin lesions. To avoid the dataset to be imbalanced and biased, benign skin lesions images and their data in the dataset have been firstly cut into 393 randomly. After carefully investigating the whole data, some data are missing for some of malignant lesions. These data have been pre-processed and deleted from the dataset. The dataset has been balanced again, leaving the dataset with the data and images of 381 malignant and benign skin lesions. Although the dataset is totally balanced after preprocessing and deleting data, the dataset is too small for analysing. So, a portion of random benign images and data is added into the dataset, leaving the ratio of benign and malignant images and data as 1.5 : 1. The dataset has been split into training, validation and testing data with the ratio of 7:2:1 respectively. These data would also be being recorded in training, validation and testing data csv files. As this project involves both traditional machine learning and deep learning, some of the data preparation methods between both would be different.

For traditional machine learning, there are some columns that are not applicable for models when processing traditional machine learning tasks. Such as 'isic_id', 'patient_id', 'lesion_id', 'iddx_full','iddx_1', 'iddx_2', 'iddx_3', 'iddx_4', 'iddx_5', 'mel_mitotic_index', 'mel_thick_mm'. 'isic_id', 'patient_id' and 'lesion_id' represents the id of the isic number, patient and lesion respectively. 'iddx_full' and 'iddx_1' indicates whether the skin lesion is benign or malignant, this data has been repeated in the 'target' column. 'iddx_2',

'iddx_3', 'iddx_4', 'iddx_5' 'mel_mitotic_index' and 'mel_thick_mm' record malignant related information if the skin is classified as malignant. These columns are not related to this traditional machine learning task, as the aim is to train the machine learning models to distinguish whether the skin is malignant or benign. These columns are dropped.

For deep learning, the images are allocated to training, validation and test folders according to training, validation and testing data csv files, and assigned into benign and malignant folders in the training, validation and test folders. The csv files have been sorted by isic_id to make sure the labels and features are aligned.

4.2 Model Application

There are six traditional machine learning and four deep learning models are utilized in this project. They are Naïve Bayes, Decision Tree, Random Forest, LGBM, CatBoost, and XGBoost for traditional machine learning models, and AlexNet, ResNet50, VGG11, and ViT for deep learning models. Data preparation and preprocessing, training, hyperparameter tuning and testing phase are all done in Google Colab. As the original dataset has 401,059 image files, much more than the limit of 100,000 image files, a reading timeout error like OSError: [Errno 5] Input/output error: occurs everytime when the image files, which are imported in the folder in Google Drive, are read by ipynbs in Google Colab. In order to overcome the problem, the image dataset is reduced according to the train-metadata-preparation.csv which is randomly selected. The reduced image dataset is put in train-image folder in Google drive for further processing. Due to time constraints and computation resources limitations, only 20 training epochs and 1 most important hyperparameter for every learning rate during hyperparameter tuning section of deep learning model are executed.

4.3 Hyperparameter tuning

As there are limited computational resources available for this project, the most important hyperparameter of models will be tuned only. It is tuned with multiple choice

and the best choice is chosen by using GridSearchCV function for traditional machine learning models. For the deep learning model, the best choice is chosen by finding the least validation when running 20 epochs.

1. Naïve Bayes

The two hyperparameters of Naïve Bayes classifier is var_smoothing and priors, var_smoothing adds a small positive value to the variance of all features, which helps to stabilize the computation of probabilities. Priors are used to set the prior probabilities of each class by users. (Rish, 2001). var_smoothing is the most important hyperparameter, so it is tuned with the choice of [1e-9, 1e-8, 1e-7, 1e-6], while 1e-9 is the best choice after using GridSearchCV function.

2. Decision Trees

The important hyperparameters of Decision Tree classifiers are random_state, max_depth, min_samples_split, min_samples_leaf, max_features and criterion. Random_state is the seed of algorithm, it ensures that the results of the algorithm can be appeared again when the algorithm re-runs. Setting the random_state as random_value can assure that the results can be appeared again. Max_depth is the maximum depth of the decision tree, it helps to control the complexity of trees. Min_samples_split represents the minimum samples required to split an internal node. It prevents overfitting as it does not allow splits on small subsets. Min_samples_leaf represents the minimum samples required in leaf node. It prevents overfitting on trees as it avoids small leaves. Max_features represents the maximum number of features, it places randomness and helps speeding up training. Criterion is the criteria that measures the quality of split. Gini impurity is usually set as the criterion. (Scikit-learn, 2024). max_depth is chosen for tuning, 5 is chosen as the best parameter from choices [None, 3, 5, 10, 20].

3. Random Forest

Random Forest classifiers have multiple important hyperparameters such as random_state, n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf and class_weight. Random_state is the seed of algorithm, it lets the results of the algorithm can be appeared again when the algorithm runs again. N_estimators represents the number of decision tree in the random forest. Class_weights are the weight of classes for handling imbalanced dataset, as the dataset of this project is a comparatively balanced dataset, class_weights is set to None, which is a default setting (Breiman, 2001). n_estimators is chosen for tuning, 50 is the best parameter within [50, 100, 200, 500].

4. XGBoost

There are multiple important hyperparameters for XGBoost such as learning_rate, max_depth, n_estimators, lambda and gamma. Learning_rate control the learning size of the model when it updates its weights during its training phase. It helps improve the generalization of the model so that the model performs better in testing sets that unseen before by the model. Lambda represents the L2 regularization on weights, it helps to reduce overfitting. Gamma represents minimum loss reduction that required for a split, it avoids the models not to be subjective (Chen & Guestrin, 2016). Learning rate is chosen for tuning, from the choices [0.01, 0.05, 0.1, 0.3], 0.05 is chosen as the best learning rate.

5. LGBM

The core hyperparameters of LGBM are boosting_type, n_estimators, learning_rate and num_leaves, max_depth, min_data_in_leaf and feature_fraction. Boosting_type represents the type of boosting algorithm, which gradient boosting decision tree (GBDT)

is set as default. Num_leaves represents the maximum of leaves in a tree, if the value is set high, it would increase model's complexity. Min_data_in_leaf is the minimum data points in a leaf. It helps reduce overfitting. Feature_fraction represents the fraction of feature per tree, it helps reduce overfitting and speed up the training phase (Ke et al., 2017). num_leaves is chosen for tuning, 32 is the best parameter within [20, 32, 64, 128].

6. CatBoost

There are some important hyperparameters for CatBoost, they are learning_rate, iterations, depth, l2_leaf_reg and random_strength. Iterations represent the number of trees. Depth represents the depth of tree. It helps to capture complex patterns. L2_leaf_reg and random_strength represents L2 regularizations on leaf values and randomness in the selection of tree structure respectively, both of them also help reduce overfitting (Prokhorenkova et al., 2018). Depth is chosen for tuning, in [4, 6, 8, 10], 8 is the best hyperparameter.

7. AlexNet

In the code of AlexNet model, there are some important hyperparameters. They are number of layers, filter size, number of filters, dropout, learning rate and activation function. There are 8 layers in AlexNet, including 5 convolutional and 3 fully connected layers. Filter size is expressed as kernel size in AlexNet, the filter size is 11x11 in the first convolutional layer, then 5x5 in the second convolutional layer, the remaining convolutional layers are 3x3. It determines the size of feature extraction, the larger size of filter responsible for capturing broader patterns, while the smaller size of filter responsible for capturing patterns in detail. Number of filters helps models to learn more features and detect diverse patterns. Dropout is applied in the first and second fully connected layers, it disables 50% of neuron randomly during training phase to prevent overfitting. The learning rate decides the size of weight updates in the training phase, if the learning rate is small, it ensures the stable learning of model in training phase, but it

may demand larger computational resources. The ReLU activation function named as nn.ReLU, its inplace is set as true, unlike the default setting as false, as it overwrites the input tensor with ReLU output, which can reduce memory usage. It is a common practice for AlexNet to optimize the time usage of computational resources (Krizhevsky, Sutskever and Hinton, 2012). Learning rate is the most important hyperparameter in deep learning model, so it is chosen to tune in all deep learning in the hyperparameter tuning. 1e-3 is chosen as the best learning rate within [1e-1, 1e-2, 1e-3, 1e-4].

8. ResNet

There are some important hyperparameters of ResNet, which are number of layers and filters, learning rate, batch size and L2 regularization. ResNet-50 is utilized in this project, 50 represents its number of layers. For the number of filters, ResNet-50 sets the first stage as 64, it doubles up in every stage. It increases capacity of features with more filters. Batch size is the number of samples in a batch in training process. L2 regularization prevents overfitting by suppressing large weights (He et al., 2016). In [1e-1, 1e-2, 1e-3, 1e-4], 1e-4 is the best learning rate.

9. VGG

The important hyperparameters of VGG are number of layers and filters, filter size, learning rate and dropout. Number of layers represents the depth of VGG, VGG11 is utilized in this project, and it has 11 layers. the configuration of VGG11 is [64, 'M', 128, 'M', 256, 256, 'M', 512, 512, 'M', 512, 512, 'M'], each M represents a max pooling layer (nn.MaxPool12d). Filters decide the number of features learned in each layer. The filters are doubled after max pooling. Dropout represents the dropout probability, and it is set to 0.5, it means that 50% of neurons have been randomly deactivated, it is used to prevent overfitting (Simonyan and Zisserman, 2015). The best learning rate is 1e-3, within the choice of [1e-1, 1e-2, 1e-3, 1e-4].

10. ViT

There are multiple important hyperparameters for XGBoost such as image size, patch size, number of layers, learning rate and number of attention heads. The image size represents the height and width of input image, it is set to 224. It means the dimension of input image is 224x224, which is one of the specialities of ViT. Patch size is the height and width of image is divided to, it is set to 16. It means the dimension of divided image is 16x16. Number of layers represents the layers of transformer encoder, it is set as 12 and is a default setting. The number of attention heads is placed in the self-attention mechanism, it is set as 12 and is a default setting (Dosovitskiy et al., 2021) 4e-5 is the best learning rate in [5e-5, 4e-5, 3e-5, 2e-5, 1e-5].

There are six traditional machine learning which are utilized in this project, including Naïve Bayes, Decision Tree, Random Forest, LGBM, CatBoost, and XGBoost as they are commonly used within Traditional Machine learning models. For deep learning, AlexNet, ResNet50, VGG11, and ViT are utilized, as these deep learning models are commonly used in ISIC dataset.

4.4 Evaluation Metrics

Confusion Matrix is an evaluation metric to measure the model performance by calculating 5 parameters: True Positive (TF), False Positive (FP), False Negative (FN), True Negative (TN) and Accuracy. True Positive (TP) reflects that Malignance (Positive) images have been classified positive correctly by the model. False Positive (FP) reflects Benign (negative) images have been classified Malignance (positive) falsely by the model. False Negative (FN) reflects Malignance (positive) images have been classified Benign (negative) falsely by the model. True Negative (TN) reflects Benign (negative) images have been classified Benign (negative) correctly by the model. Accuracy means the accuracy of the model classification. Precision and Recall are evaluation metrics which used the parameters of Confusion Matrix as further evaluation of the models. Precision measures the ability of the model to detect positive images correctly. While

recall measures the ability of the model to detect all the actual positive images. F1 score components are Precision and Recall. It is useful to detect imbalanced dataset or predict bias as it combines the performance of Precision and Recall, if one of them is too low, it would largely affect the score.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Prescision = \frac{TP}{TP + FP}$$

$$Recall/Sensitivity = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

Receiver Operation Characteristic Curve (ROC) reflects the models' performance of different decision thresholds. X-axis represents the False Positive Rate (FPR), while Y-axis represents True Positive Rate (TPR). Area Under the Curve (AUC) represents the area under the ROC curve. It reflects the performance of the model when detecting a specific class. The score is within 0 to 1. When the score is 0.5, it reflects the model is guessing randomly, while the score is closer to 1, the model has a better performance.

Chapter 5 - Evaluation

5.1 Analytical and Reflective Discussion on results

After implementing traditional machine learning and deep learning models to ISIC 2024 dataset, the performances of all models are displayed in evaluation metrics, which are shown below:

Model	Hyperparameter Tuning	Hyperparameter Tuning Trial	Best Hyperparameter	Test Accuracy	ROC score	Unnormalized Confusion Matrix	Normalized Confusion Matrix	Precision	Recall	F1 Score
AlexNet	learning_rate	[1e-1, 1e-2, 1e-3, 1e-4]	0.001	0.562	0.581	[[36 22] [20 18]]	[[0.62 0.38] [0.53 0.47]]	0.546	0.547	0.547
ResNet50	learning_rate	[1e-1, 1e-2, 1e-3, 1e-4]	0.0001	0.625	0.613	[[47 11] [25 13]]	[[0.81 0.19] [0.66 0.34]]	0.597	0.576	0.571
VGG11	learning_rate	[1e-1, 1e-2, 1e-3, 1e-4]	0.001	0.615	0.587	[[42 16] [21 17]]	[[0.72 0.28] [0.55 0.45]]	0.591	0.586	0.587
ViT	learning_rate	[5e-5, 4e-5, 3e-5, 2e-5, 1e-5]	4e-5	0.583	0.634	[[43 15] [25 13]]	[[0.74 0.26] [0.66 0.34]]	0.548	0.542	0.538

Gaussian Naive Bayes	var_smoothing	[1e-9, 1e-8, 1e-7, 1e-6]	1e-9	0.812	0.888	[[50 8] [10 28]]	[[0.86 0.14] [0.26 0.74]]	0.806	0.799	0.802
Decision Tree	max_depth	[None, 3, 5, 10, 20]	5	0.792	0.719	[[50 8] [12 26]]	[[0.86 0.14] [0.32 0.68]]	0.786	0.773	0.778
Random Forest	n_estimators	[50, 100, 200, 500]	50	0.844	0.911	[[48 10] [5 33]]	[[0.83 0.17] [0.13 0.87]]	0.837	0.848	0.840
XGBoost	learning_rate	[0.01, 0.05, 0.1, 0.3]	0.05	0.885	0.928	[[52 6] [5 33]]	[[0.9 0.1] [0.13 0.87]]	0.879	0.882	0.881
LGBM	num_leaves	[20, 32, 64, 128]	32	0.865	0.921	[[50 8] [5 33]]	[[0.86 0.14] [0.13 0.87]]	0.857	0.865	0.860
CatBoost	depth	[4, 6, 8, 10]	8	0.844	0.934	[[33 5] [7 32]]	[[0.87 0.13] [0.18 0.82]]	0.845	0.844	0.844

Table 5.1: Evaluation metrics of 6 traditional machine learning and 4 deep learning models

Gradient boosting algorithms (LGBM, CatBoost and XGBoost) and Random Forest from traditional machine learning models perform comparatively better than other models. XGBoost outperformed from all models with its achievement of highest test accuracy of 88.5%, F1 score of 88.1%, precision of 87.9% and recall of 88.2%, while XGBoost is followed closely by LGBM with a very slightly difference in most of the metrics. Deep learning models (AlexNet, ResNet50, VGG11, and ViT) underperformed compared to traditional machine learning models.

The models are tuned with the hyperparameters which have a greatest impact on results during the training phase. For tree-based models, the hyperparameters tuned are mostly related to trees such as max depth of tree, number of leaves and n_estimators (number of trees generated in random forest). The greatest impact of hyperparameter on results of deep learning models are learning rate, and it has been tuned in the training phase of deep learning models. ResNet50 displayed the best performance within deep learning models, with its 62.5% test accuracy, 61.3% ROC score and 57.1% F1 score. AlexNet performed the worst within traditional machine learning and deep learning models, with its 56.2% test accuracy, 58.1% ROC score, 54.6% precision of evaluation metrics are the lowest.

Test accuracy is a basic performance metric, it can evaluate the effectiveness of the model. Traditional machine learning with Gradient boosting algorithms and Random Forest performed comparatively well than the remaining traditional machine learning and deep learning models in test accuracy. The test accuracy of Traditional machine learning with Gradient boosting algorithms and Random Forest are in the range of 84.4% to 88.5%, while the remaining traditional machine learning models have test accuracy ranging from 79.2% to 81.2%. On the other hand, the test accuracy of deep learning models is under 62.5%. The high accuracy scores of these gradient boosting algorithms and Random Forest reflect that they can distinguish between the features of benign and malignant skin lesions even in small volume of training data.

For the confusion matrix, XGBoost performed the best within the models. It classified 90% of benign (class 0) skin lesions and 87% malignant (class 1) skin lesions correctly. Other machine learning models performed impressively in confusion matrix, they can

classify both types of skin lesions in a decent percentage. On the other hand, deep learning models cannot classify class 1 correctly, all of them cannot classify class 1 correctly for a half. It reflects the instability of deep learning models when given limited data in binary classification tasks.

For F1 score, precision and recall, Gradient boosting algorithms and Random Forest scored within 84.0%-88.1%. They performed comparatively better then deep learning models (53.8%-58.7%) and maintained a balance between precision and recall which proved by their stable performance on F1 score. Other traditional machine learning models (77.8%-80.2%) performed steadily, their scores on precision and recall are nearly perfectly balanced. On the other hand, deep learning models although maintained a certain balance on precision and recall, they all scored a low precision and recall. VGG11 performed best within deep learning models, it scores in precision and recall of 59.1% and 58.6% respectively, while its F1 score is 58.7%. Although the ratio of benign and malignant data and images are in 1.5:1 ratio, all models achieved a balance on precision and recall when analyzing this dataset, this dataset seems to be a balanced dataset that may only have a minor bias.

ROC score measures the robustness of the models. Random forest and gradient boosting algorithms performing well in ROC score, ranging from 91.0%-93.4%. It reflects these models are highly consistent in their performance. On the other hand, deep learning models scored in the range of 58.1% to 63.4%, where ViT performed the best, scoring 63.4%.

5.2 Research limitations

The major limitation is computational resource limitations. The only platform utilized in this project is Google Colab for processing the models. Instead of CPU, it only provides very limited GPU hours usage per day. It does not have sufficient time to run too many epochs as Google Colab costed numerous hours to execute 20 training epochs for every learning rate during hyperparameter tuning section of deep learning model when using CPU. In this project, there are 4 deep learning models with 4 to 5 learning rates to test and to find the best learning rate for hyperparameter tuning section in each model. In

contrast, researchers in other academic papers would do at least 100 to 150 epochs for every model to find the best training result for their study. This leads to their models' performance are far better.

Another limitation is the number of valid malignant samples being too small. After data preprocessing, the valid malignant data and images are reduced to 381. To prevent the dataset from being unbalanced and biased, benign samples should not be significantly more than malignant samples, this leads to a smaller overall dataset which affects the effectiveness of model training.

Chapter 6 - Conclusion

6.1 Conclusion

This project evaluated and compared the performance of traditional machine learning and deep learning models for the classification of skin lesions using a distinctive ISIC 2024 SLICE-3D dataset. The dataset contains, in additional to images dataset, metadata of physical lesion parameters, such as lesion size in mm lesion colour and contrast and irregularity, and 3D location data, which is directly derived from 3D Total Body Photography (3D-TBP). It provides a good opportunity to examine how traditional machine learning and deep learning algorithms perform in medical dataset which structured features and image inputs. On the other hand, the study was limited by insufficient malignant lesions sample and computational capacity and resources using Colab.

In the preliminary research, the results demonstrated that traditional machine learning models especially gradient boosting algorithms (Test accuracy: 84.4%-88.5%, F1 score: 84.4%-88.1%), such as XGBoost, LGBM, and CatBoost outperformed deep learning models (Test accuracy: 56.2%-62.5%, F1 score: 53.8%-58.7. Traditional machine learning models scored higher in 79.2%-84.4% accuracy, 77.8%-84.0% F1 score, they showed their better robustness in their high ROC-AUC score. Throughout, XGBoost displayed the best performance (Test accuracy: 88.5%, F1 score: 88.1%) throughout the models tested, which proves that it can extract and capture features effectively in small medical datasets.

In contrast, deep learning models such as ResNet50, AlexNet, VGG11, and ViT are underperformed. They cannot classify malignant images correctly, as deep learning models may not perform well in small datasets and insufficient training epochs. The performances of deep learning models are affected by computational resource constraints. The limited of GPU time leads to shorter training epochs (20 epochs), which is not adequate for complex deep learning architectures to learn effectively.

The number of valid malignant samples (381) are too little for training deep learning models, which causes overfitting on them. This maybe the reason of the difference of performance between deep learning and traditional machine learning models.

The project demonstrated the strengths of traditional machine learning models, especially gradient boosting algorithms, in classification tasks using small medical datasets with metadata of physical lesion parameters effectively. But it also displays some limitations of insufficient malignant samples and computational resources which need to be resolved.

This project achieves its objectives completely through several key steps. First, it utilizes the ISIC 2024 - Skin Cancer Detection with 3D-TBP SLICE-3D dataset, which includes both images and metadata of physical lesion parameters. The dataset is preprocessed and balanced to maintain a 1.5:1 ratio of benign to malignant samples. Various traditional machine learning models—Naïve Bayes, Decision Tree, Random Forest, XGBoost, LightGBM, and CatBoost—are selected for implementation. In addition, deep learning models including ResNet, AlexNet, VGG, and Vision Transformer (ViT) are employed. Performance is evaluated using relevant evaluation metrics such as test accuracy, ROC-AUC score, precision, recall and F1 score to compare traditional and deep learning approaches. Finally, insights are provided to determine which modelling approach is more suitable for skin lesion classification, offering a foundation for future research.

Thus, this project aimed to evaluate and compare different Machine Learning models for classifying skin lesions using the ISIC dataset, which includes both lesion metadata and image data, and this aim has now been successfully completed.

6.2 Future Work Recommendations

Gradient boosting algorithms (LGBM, CatBoost and XGBoost) of traditional machine learning models are good at feature extraction. It can calculate and list the feature importance of every features. By finding the occurrence of features, some important

features can be extracted and using only these features to let the model analyse, it may enhance models' performance as this approach helps to filter out irrelevant features, which may improve the effectiveness of the training phase. All features with significant feature importance can joint together and form as combinations to test and find the best diagnostic features for classification, similar to Bistroń and Piotrowski (2022)'s work, which may improve the classification accuracy of detecting skin cancer.

Using pre-trained deep learning models as feature extractors, and combines the features extracted and fed to traditional machine learning classifiers (Shakya, Patel & Joshi, 2025), may also a good approach to obtain impressive results.

There are different architectures of ResNet and VGG such as ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152 (He et al., 2016), and VGG11, VGG13, VGG16 and VGG19 (Simonyan and Zisserman, 2015). Apart from ResNet50 and VGG11 which utilized in this dataset, the remaining architectures are potential models to explore their performance in medical datasets, to find out whether they can enhance the performance on this project. Combining these deep learning architectures as Thwin and Park's (2024) work is also a promising option to achieve better results.

Last but not least, better computational resources and time are also recommended to continue this project in the future, as running sufficient epochs in training phase of deep learning models is crucially important to improve the performance of models.

References

Anilkumar, B., Velaga, S.M. and Devi, A.A., 2019. Text and Non Text Scene Image Classification for Visually Impaired Through Alexnet Transfer Learning Model. International Journal of Recent Technology and Engineering (IJRTE), 8(1), pp.1125–1129.

Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J.R., Grabska-s, A., Taylor, K.R., Assael, Y., Jumper, J., Kohli, P. and Kelley, D.R., 2021. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10), pp.1196–1203.

Bistroń, M. and Piotrowski, Z. (2022) 'Comparison of Machine Learning Algorithms Used for Skin Cancer Diagnosis', *Applied Sciences*, 12(19), p. 9960. doi: 10.3390/app12199960.

Bozyel, S., Şimşek, E., Koçyiğit Burunkaya, D., Güler, A., Korkmaz, Y., Şeker, M., Ertürk, M. and Keser, N., 2024. Artificial intelligence-based clinical decision support systems in cardiovascular diseases. *Anatolian Journal of Cardiology*, 28(2), pp.74–86.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, **24**(2), pp.123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001) *Machine Learning*, 45(1), pp. 5–32. doi:10.1023/a:1010933404324.

Chen, R., Zhou, L., Xiong, C., Xu, H., Zhang, Z., He, X., Dong, Q. and Wang, C., (2023). *Islanding detection method for microgrids based on CatBoost*. *Frontiers in Energy Research*, 10, p.1016754. doi:10.3389/fenrg.2022.1016754.

Chen, T. and Guestrin, C. (2016) 'XGBoost', *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. doi:10.1145/2939672.2939785.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N., (2021). *An image is worth 16x16 words: Transformers for image recognition at scale*. arXiv preprint arXiv:2010.11929.

Hafeez, M.A., Rashid, M., Tariq, H., Abideen, Z.U., Alotaibi, S.S. and Sinky, M.H., (2021). *Performance improvement of decision tree: A robust classifier using tabu search algorithm*. *Applied Sciences*, 11(15), p.6728. doi: 10.3390/app11156728.

He, K., Zhang, X., Ren, S. and Sun, J. (2016) 'Deep residual learning for image recognition', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. doi:10.1109/CVPR.2016.90.

Ho, T.K., 1995. *Random decision forests*. In: Proceedings of the Third International Conference on Document Analysis and Recognition. Montreal, QC, Canada: IEEE, pp.278–282. https://doi.org/10.1109/ICDAR.1995.598994.

Huo, Y., Jin, K., Cai, J., Xiong, H. and Pang, J. (2023) *Vision Transformer (ViT)-based applications in image classification*. In: 2023 IEEE 9th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC) and IEEE International Conference on

Intelligent Data and Security (IDS). New York, NY, USA, 1–3 May 2023. New York: IEEE, pp. 135–140. doi: 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00033

Ioffe, S. and Szegedy, C., 2015. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp.448–456.

International Skin Imaging Collaboration. (2024) *ISIC 2024 Challenge Dataset*. Available at: https://challenge2024.isic-archive.com/ (Accessed: 5 Mar 2025).

International Skin Imaging Collaboration (ISIC), 2024. *SLICE-3D 2024 Challenge Dataset*. [online] https://doi.org/10.34970/2024-slice-3d [Accessed 5 Mar 2025].

International Skin Imaging Collaboration. *SLICE-3D 2024 Challenge Dataset*. International Skin Imaging Collaboration https://doi.org/10.34970/2024-slice-3d (2024).

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q. & Liu, T.-Y. (2017), LightGBM: A Highly Efficient Gradient Boosting Decision Tree, in I. Guyon; U. V. Luxburg; S. Bengio; H. Wallach; R. Fergus; S. Vishwanathan & R. Garnett, ed., *'Advances in Neural Information Processing Systems 30'* , Curran Associates, Inc., , pp. 3146--3154.

Khan, M.Y., Qayoom, A., Nizami, M.S., Siddiqui, M.S., Wasi, S. and Raazi, S.M.K.-u.-R., (2021). Automated prediction of good dictionary examples (GDEX): A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques. Complexity, 2021. doi: 10.1155/2021/2553199

Kiran, I., Siddique, M.Z., Butt, A.U.R., Mudassir, A.I., Qadir, M.A. and Munir, S., (2021). *Towards skin cancer classification using machine learning and deep learning algorithms: A comparison*. International Journal of Innovations in Science & Technology, 3(4), pp.110–118. https://doi.org/10.33411/IJIST/2021030508.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) 'ImageNet classification with deep convolutional neural networks', in Pereira, F., Burges, C.J.C., Bottou, L. and Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25 (NIPS 2012)*. Curran Associates, Inc., pp. 1097–1105.

Kumar, A., Sharma, R., & Gupta, P. (2024). Advancements and applications of artificial intelligence in cardiology: Current trends and future prospects. *Journal of Cardiovascular Medicine*, 19(3), pp.123–135.

Kundu, N. (2023) 'Exploring ResNet50: An in-depth look at the model architecture and code implementation', *Medium*, 23 January. Available at: https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f (Accessed: 27 April 2025).

Mahbod, A., Schaefer, G., Wang, C., Ecker, R. & Ellinger, I., (2019). *Skin lesion classification using hybrid deep neural networks*. [online] arXiv. https://arxiv.org/abs/1702.08434v2 [Accessed 6 Apr 2025].

Maron, M.E., (1961). Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3), pp.404–417. https://doi.org/10.1145/321075.321084

Morgan, J.N. and Sonquist, J.A., (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58(302), pp.415–434.

Naeem, M.A., Yang, S., Sharif, A., Saleem, M.A. and Sharif, M.I., (2024). *Vision Transformer for skin cancer identification based on contrastive learning and adaptive-scale fragmentation*. [online] Research Square. https://doi.org/10.21203/rs.3.rs-4271003/v1 .

Nancy, V.A.O., Prabhavathy, P., Arya, M.S. and Ahamed, B.S., (2023). *Comparative study and analysis on skin cancer detection using machine learning and deep learning algorithms*. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-16422-6

Natha, P. and RajaRajeswari, P., 2024. *Advancing skin cancer prediction using ensemble models*. Computers, 13(7), p.157. https://doi.org/10.3390/computers13070157.

Nlztrk (2024) *My competition summary - ISIC 2024*. Medium. Available at: https://medium.com/@nlztrk/my-competition-summary-isic-2024-825ab1b82711 (Accessed: 10 April 2025).

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V. and Gulin, A., 2018. *CatBoost: unbiased boosting with categorical features*. arXiv preprint arXiv:1706.09516.

Rath, S.R. (2021) *Implementing VGG11 from scratch using PyTorch*. Available at: https://debuggercafe.com/implementing-vgg11-from-scratch-using-pytorch/ (Accessed: 29 April 2025).

Rayner, J.E., Laino, A.M., Nufer, K.L., Adams, L., Raphael, A.P., Menzies, S.W. and Soyer, H.P., 2018. *Clinical perspective of 3D total body photography for early detection and screening of melanoma*. Expert Review of Medical Devices, 15(5), pp.337–346.

Rish, Irina. (2001). *An Empirical Study of the Naïve Bayes Classifier*. IJCAI 2001 Work Empir Methods Artif Intell. 3.

Scikit-learn, 2024. sklearn.tree.DecisionTreeClassifier. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html [Accessed 20 Mar. 2025]

Shakya, M., Patel, R. & Joshi, S., (2025). *A comprehensive analysis of deep learning and transfer learning techniques for skin cancer classification*. Scientific Reports, 15(1), p.4633. https://doi.org/10.1038/s41598-024-82241-w

Skin Cancer Foundation, 2024. *Skin Cancer Facts & Statistics*. [online] Available at: https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/ [Accessed 21 Mar. 2025].

Simonyan, K. and Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Thwin, S.M. and Park, H.-S., (2024). *Skin lesion classification using a deep ensemble model*. *Applied Sciences*, 14(13), p.5599. https://doi.org/10.3390/app14135599.

Quinlan, J.R. (1986) ‘Induction of Decision Trees’, *Machine Learning*, 1(1), pp. 81–106. doi:10.1007/bf00116251.

World Health Organization (WHO), 2024. *Cancer*. [online] Available at: https://www.who.int/news-room/fact-sheets/detail/cancer [Accessed 11 Mar. 2025].

Yao, X., Fu, X. and Zong, C., (2022). *Short-term load forecasting method based on feature preference strategy and LightGBM-XGboost*. IEEE Access, 10, pp.75257–75268. doi:10.1109/ACCESS.2022.3192011.

Appendix A

Feasibility study powerpoint link:

https://stummuac-my.sharepoint.com/:p:/r/personal/22454220_stu_mmu_ac_uk/Documents/Documents/22454220%20Bo%20Kwok%20RM%20-%20Slide%20Deck.pptx?d=w64645ec3e6a04b53b20af285448d5a07&csf=1&web=1&e=9Tzxtq