A REPORT ON

Exploring Various Machine And Deep Learning Models in HealthCare Dataset

By

Name of the student	Enrolment/Registration No.
YASH AGRAWAL	230745

Mentored by:
Dr.Shilpa Mahajan
(Assistant Professor)

Prepared in the partial fulfillment of the

Practice School II Course

At

BML Munjal University, Guragon, Haryana,122413

BML MUNJAL UNIVERSITY

(September, 2025)

Certificate of authenticity

CERTIFICATE

This is to certify that Practice School Project of Yash Agrawal titled Exploring Various Machine And Deep Learning Model In HealthCare Dataset is an original work and that this work has not been submitted anywhere in any form. Indebtedness to other works/publications has been duly acknowledged at relevant places. The project work was carried during 2/6/25 to 27/7/25 under the guidance of Dr. Shilpa Mahajan.

Signature of Supervisor:

Name: Dr. Shilpa Mahajan

Designation: Assistant Professor

JOINING REPORT

Date:23/06/25

Name of the Student	**Yash Agrawal**
Name and Address of the Practice School – II Station	**BML Munjal University**
Date of Joining PS-II station as per offer letter	02/06/25
Actual date of reporting to PS-II station	05/06/25
Department Allocated	Thesis in A **Artificial Intelligence and Machine Learning**
Name and Designation of the Industry Guide/ Industry Mentor for the Project	Dr. Shilpa Mahajan – Assistant Professor
Industry Mentor Contact No.	9882450155
Industry Mentor E-mail Address (Compulsory)	shilpa.mahajan@bmu.edu.in

Acknowledgements

I would like to thank every single person who helped me during my Practice School-II project. Their guidance, encouragement, and advice were essential to the completion of this report.

To begin with, I would like to thank Prof. Shyam Menon, Vice Chancellor, BML Munjal University, and Prof. Maneek Kumar, Dean of the School of Engineering and Technology. They provided me with the opportunity to work on this project and created such a valuable learning platform.

I would also like to thank Dr. Shilpa Mahajan, my faculty mentor, for ongoing academic support, useful tips, and follow-up. All this guidance seriously impacted the course of my work.

Finally, I would like to acknowledge and appreciate each of the members, staff, and other professionals from organization and others. Their information, support, and experience were required to complete her project.

This project has been a great learning experience, and I am thankful to all who took part in it.

Abstract

This project explores the use of deep learning in the assistance of early disease diagnosis from medical images. We experimented and trained various types of convolutional neural network models on three well-known datasets Messidor-2 for diabetic retinopathy detection, and HAM10000 for skin lesion classification. We used pre-trained models like ResNet50 and MobileNetV2, and our own models. In certain cases, we augmented metadata like age, gender, and lesion location with imaging features to improve results. We examined special methods like attention mechanisms and dual-input models to enhance prediction performance and comprehensible. The models were also compared on metrics such as accuracy, precision, recall, and F1-score. In this paper, we demonstrate how AI may be an effective utility in assisting doctors by providing quick and certain second opinions, principally in areas of limited access to medicine.

It was conducted under the supervision of Dr. Shilpa Mahajan at BML Munjal University under the Practice School-II initiative.

Table of Contents

Plan of The Thesis

The thesis project was conducted under **Department of Computer Science, School of Engineering and Technology (SoET)** of **BML Munjal University**, as part of the **Practice School-II** program. The project was conducted under the guidance and supervision of **Shilpa Mahajan**.

Thesis Duration:

Start Date: 2nd June, 2025

End Date: 27th July, 2025

This. project seeks to establish the optimum deep learning model to apply in the classification of medical images with special emphasis on skin and retinal conditions. The work is based on three well-documented datasets —Messidor-2 for diabetic retinopathy (Decenciere et al., 2014), and HAM10000 for skin lesion classification (Tschandl et al., 2018) — to enable comparison of model performance on a broad spectrum of medical imaging cases.

The development was done following a systematic approach:

Problem & Goal
Explored the need for AI in medical diagnosis. Aim: find the best model for classifying medical images.

Datasets Used
Worked with HAM10000, Messidor, and Skin Cancer MNIST. Cleaned and prepped each for training.

Model Approach
Tested CNNs, ResNet, MobileNet , Inception and Efficient models to see which performs best on different datasets.

Implementation
Used TensorFlow/Keras. Applied techniques like augmentation and early stopping to improve results.

Evaluation and Conclusion
Compared models using accuracy, F1-score, etc. Visualized results with graphs and confusion matrices. Identified the most reliable model. Suggested improvements like better data and future real-world use.

Introduction

Healthcare lies at the heart of a nation's development as a whole, and in a nation as highly populated as India, it is paramount. The scale of the issues is big, and yet over the past few decades, India's healthcare industry has come up by leaps and bounds. Improved hospitals now exist, along with higher availability of medicine, swifter and correct diagnosis, and more incorporation of technology in both treatment and care of patients.

One of the extremely significant fields in medicine is medical diagnosis, that is, diagnosing illness so that at the due time, effective treatment could be provided to patients. In India, high-level diagnostic centers and qualified professionals are concentrated in cities. The urban-rural disparity typically results in delay or, in certain cases, incorrect diagnosis of patients in interior areas. The situation becomes more critical in conditions requiring imaging studies like pneumonia (chest X-rays), diabetes-induced retinopathy (scans of the eye), or skin cancer (dermoscopic pictures).

In recent years, artificial intelligence (AI) has begun to revolutionize the world of medicine by offering new solutions. AI, in particular by means of deep learning methods such as convolutional neural networks (CNNs), enables computers to read medical pictures and spot patterns that could be a sign of illness. AI systems, in this case, aren't meant to take the place of doctors, but as a quick, trustworthy "second opinion," anywhere, anytime.

The case of India is encouraging because such AI-driven diagnostic devices can be installed in primary health centers or even in mobile phones. In this way, individuals in rural villages can obtain preliminary screening without covering long miles to get to cities. In life-and-death cases, early detection can be a lifesaver. Furthermore, AI can also decrease substantially the work of doctors who are, in many cases, overworked. The AI systems will be able to promptly analyze big data of scans, highlight suspicious cases, and leave doctors to direct attention at priority patients. Hospitals, startups, and government initiatives today are actively seeking how artificial intelligence will help them create more intelligent, swifter, and more equal healthcare for all.

Overall, this pairing of technology and medicine, especially AI, is introducing a new ERA of possibility in India. This revolution is more than about acceleration of diagnosis—it is about more accurate, more affordable, and more equal care to every person, no matter what place they inhabit.

Background

Medical imaging has, over the past few decades, emerged as an essential modality for early and accurate diagnosis of disease. Ranging from the detection of malignant skin lesions to the detection of diabetic retinopathy on the retina, medical images represent the center of clinical decision-making. But accurate interpretation of these images demands enormous expertise and is often cumbersome and prone to human mistakes.

That's where convolutional neural networks (CNNs) and deep learning come in. They've been shown to be very promising at making good predictions from complex medical images. But with as many architectures and techniques as there are nowadays, it's normally hard to know exactly which model is most suitable for a particular medical condition or dataset.

The objective of this project is to compare and discuss the performance of different deep learning models on three large medical image datasets:

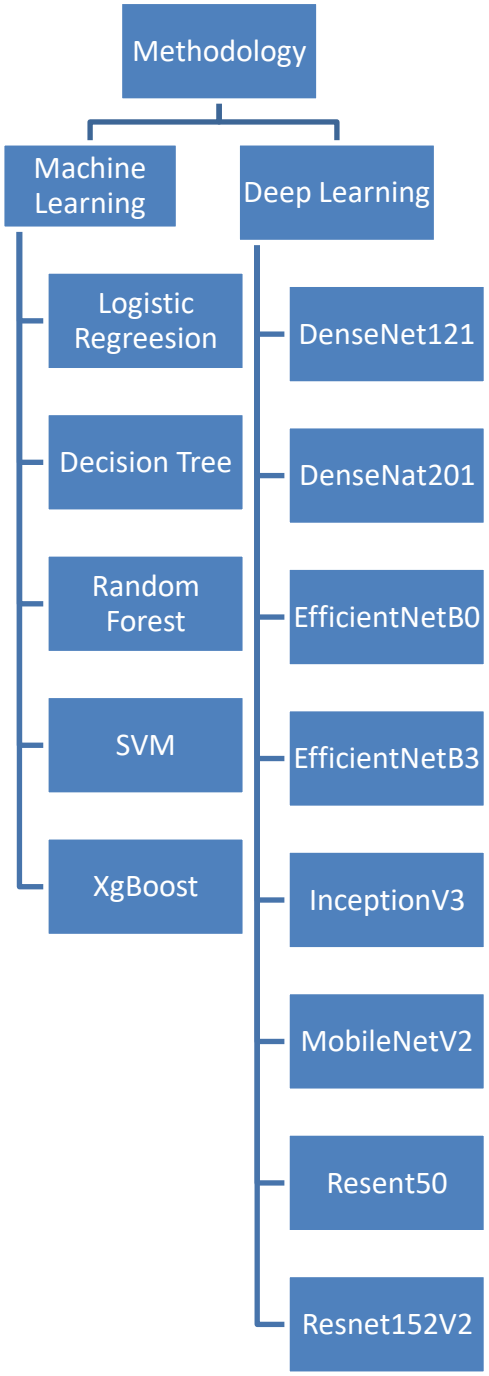
- HAM10000 dataset (Tschandl et al., 2018) consisting of dermatoscopic skin lesion images belonging to various types
- Skin Cancer MNIST (ISIC), a widely used skin cancer classification dataset
- Messidor-2 database (Decenciere et al., 2014), a retinal image database widely employed in detection of diabetic retinopathy studies.

Every dataset is a unique medical problem. Through experimentation on models such as ResNet50, MobileNetV2, and more sophisticated methods such as dual-input models (image + metadata) and attention-based Multiple Instance Learning (MIL), we seek to identify which methods succeed and why.

This research is not merely concerned with improving accuracy but also finding models that are effective, robust, and interpretable enough to apply in real-world healthcare settings. The ultimate goal is to assist in the development of improved diagnostic devices that will allow clinicians to make quicker and more accurate decisions, which ultimately lead to improved patient outcomes.

Methodology

In this chapter , I discussed the Machine Learning and deep learning techniques to determine which of the two works the best for medical image classification.



• Logistic Regression

Logistic Regression was my starting point since it gives a simple yet efficient baseline for classification. It gives an estimate of the probability that an input will belong to a specific class and can be applied to both binary and multi-class classification. In this project, I used it to classify the medical images by converting the data into numerical features. One advantage of Logistic Regression is that it is simple to interpret since the coefficients directly indicate the significance of each feature. Though it is not efficient with complex and non-linear data, which is common for medical images, for better efficiency, I used feature scaling and regularization. These techniques helped to contain overfitting and stabilized the training process. Though Logistic Regression is less robust compared to other models, it gave a baseline against which the improvements offered by the more sophisticated methods could be gauged.

• Decision Tree

Decision Trees were the second option since they can learn non-linear relationships between features. Decision Trees divide the dataset into smaller homogeneous subsets according to decision rules at each node. In medical image classification, what the model learns are feature or pixel value patterns or extracted features. Decision Trees are simple to visualize and interpret, hence explainable AI in medicine would be desirable. They are prone to overfitting, however, particularly if the tree is too deep. To prevent this, I set a limit to the maximum depth and pruned the tree to find the balance between complexity and generalization. Not ideal, but Decision Trees are a good starting point to learn the data structure prior to ensemble methods.

• Random Forest

Random Forest generalizes Decision Trees by combining many of them into an ensemble. Each tree is trained on a random subset of data and feature set, and their predictions are combined by majority vote. This both constrains overfitting and increases robustness. Random Forest helped me in my work to deal with the high medical image variability by averaging results over many decision paths. It also provided feature importance scores, which informed which variables were most accountable for the predictions. Training takes longer than for a single tree, but the improvement in performance is dramatic. I used hyperparameter tuning to determine the number of trees, depth, and split metric. The final model needed to balance accuracy and interpretability and was therefore a safe choice for structured medical data.

• Support Vector Machine (SVM)

SVM is known for its ability to deal with high-dimensional data and is thus best suited for image classification problems. It is based on the idea of the best hyperplane with the largest margin that maximally distinguishes the classes. I tried various kernel functions such as linear, polynomial, and RBF to achieve non-linear boundaries. SVMs are computationally expensive for big databases but should be highly accurate if properly tuned. To make the model efficient, I carried out dimensionality reduction and feature scaling prior to training. One of the challenges was imbalanced data, and I addresses this by applying class weights. SVM was a tough competitor among the conventional machine learning models, particularly when the data was well preprocessed.

• XGBoost

XGBoost is a high-performance gradient boosting algorithm that builds an ensemble of trees sequentially. Each additional tree attempts to reduce the errors of the previous ones. It is efficient, scalable, and is known to deliver best-in-class performance on structured data problems. In my project, XGBoost learned complex feature interactions in medical images and metadata. I used grid search for parameter tuning like learning rate, max depth, and number of estimators. Its robust tolerance to missing values and imbalanced data made it suitable for healthcare. Compared to Random Forest, XGBoost provided better accuracy and faster convergence. The only major limitation is that it needs careful parameter tuning, but after optimization, it delivered stable performance on multiple evaluation metrics.

• **Hyperparameter Tuning**
To ensure model comparison was fair, I conducted hyperparameter tuning for all the models. This included identifying the best set of parameters to ensure maximum performance. For simple models like Logistic Regression and SVM, tuning meant changing parameters like regularization strength and kernel type. For ensemble models like Random Forest and XGBoost, it meant changing the number of estimators, max depth, and learning rate. I used techniques like Grid Search and Random Search with cross-validation to compare different sets of parameters. Hyperparameter tuning enhanced the accuracy and generalizability of each model significantly, so they were being compared at their best.

• **ResNet50**
On transitioning to deep learning, I employed ResNet50, which is a convolutional neural network with residual connections. Skip connections rectified the vanishing gradient issue, enabling the network to train efficiently even with 50 layers. In medical images, this was useful as it facilitated the extraction of intricate visual features. Transfer learning was employed by pre-training the model through weights trained on ImageNet and subsequently fine-tuning the final few layers using my dataset. Data augmentation operations like rotation, flipping, and zooming were employed to enhance generalization. Early stopping and dropout were employed to avoid overfitting. ResNet50 was useful and yielded stable accuracy while preserving training stability.

• **MobileNetV2**
I chose MobileNetV2 because it is light and efficient and well-suited for resource-constrained environments. It applies depthwise separable convolutions and inverted residuals, reducing computations without accuracy loss. In the project, MobileNetV2 offered an efficient alternative to more computationally expensive networks like ResNet50. It was especially valuable in situations where efficiency was paramount, like deployment on mobile or edge devices. I applied transfer learning with pre-trained ImageNet weights and fine-tuned the last layers. Though smaller in size, MobileNetV2 performed as well as large networks when trained using data augmentation. Its efficiency and speed made it a good choice for low-resource or real-time applications.

• **EfficientNetB0**
EfficientNetB0 is one of the EfficientNet models that implement compound scaling to scale width, depth, and resolution for optimal performance. EfficientNet does not scale randomly like other networks but enhances accuracy with fewer parameters. EfficientNetB0 was used in this study as a balance between accuracy and efficiency. Using transfer learning, I also fine-tuned the model on the dataset with augmentation and regularization for better generalization. It trained relatively fast compared to ResNet50 and had good accuracy, making it one of the best-balanced models in this study. EfficientNetB0 confirmed that high performance does not always come hand in hand with very deep or heavy models, which is crucial in actual real-world healthcare applications.

• **Inception V3**
Inception V3 is a convolutional neural network architecture from Google that extends the Inception (Google Net) model to attain high accuracy while using fewer computations. Inception V3 applies Inception modules that use multiple convolution filters of various sizes in parallel, allowing the network to extract features at multiple scales compactly. The model incorporates methods like factorized convolutions that divide larger convolutions into smaller ones for lower parameter usage and also makes use of auxiliary classifiers for enhanced gradient flow during learning. Techniques like label smoothing for regularization avoid overfitting and render Inception V3 very effective for applications involving image classification and feature learning. Inception V3 is popular in computer vision applications because it balances the use of depth, efficiency, and performance.

• DenseNet121

DenseNet121 is a convolutional neural network architecture belonging to the DenseNet class that links each layer to every other layer in a feed-forward manner. Within this architecture, the feature maps of all previous layers are concatenated** and utilized as inputs for each of the following layers, enhancing feature reuse, preventing the vanishing gradient problem, and decreasing the number of parameters relative to traditional CNNs. DenseNet121 comprises 121 layers, such as dense blocks and transition layers that downsample feature maps without losing information. It is most commonly used for image classification, medical image analysis, and transfer learning applications because it's efficient, has strong gradient flow, and high accuracy.

• DenseNet201

DenseNet201 is a dense convolutional neural network from the DenseNet series that adheres to the dense connectivity principle, in which every layer takes inputs from all previous layers and shares its own feature maps with all following layers. This structure promotes **feature reuse, fortifies gradient flow, and decreases parameters to those of standard architectures. DenseNet201 has 201 layers, which makes it deeper and stronger compared to DenseNet121, with several dense blocks with transition layers in between for downsampling. It has high accuracy on big databases such as ImageNet and is frequently utilized in tasks like medical image classification, object recognition, and transfer learning, where efficient learning and diverse feature extraction are necessary.

• ResNet152V2

ResNet152V2 is a highly deep convolutional neural network belonging to the ResNet (Residual Network) series that adds residual connections to simplify the training of very deep models. In contrast to normal networks, it employs skip connections that add the input of a layer to its output directly, assisting in overcoming the vanishing gradient problem and enabling networks with more than 100 layers to train efficiently. The "V2" version enhances the base ResNet by using **pre-activation of batch normalization and ReLU before convolutions**, which regularizes training and generalizes better. With 152 layers, ResNet152V2 is among the deepest ResNet models, with robust performance on image classification tasks such as ImageNet, and is extensively used for transfer learning, feature extraction, and computer vision applications needing highly robust and accurate models..

• EfficientNetB3

EfficientNetB3 is a Google-developed convolutional neural network from the EfficientNet family that was built for high-accuracy performance while remaining computationally efficient. It applies a compound scaling technique that equally scales the network's depth, width, and resolution for improved performance while not heavily increasing parameters or computation. A continuation of the Mobile Inverted Bottleneck Convolution (MBConv) blocks, the model also integrates swish activation and squeeze-and-excitation modules for enhanced feature representation. Higher accuracy has been achieved by it for large image datasets such as ImageNet when compared to other traditional models, and it is accomplished through that model's use of fewer parameters and FLOPS. This makes it perfect for use for image classification, transfer learning, and feature extractive applications.

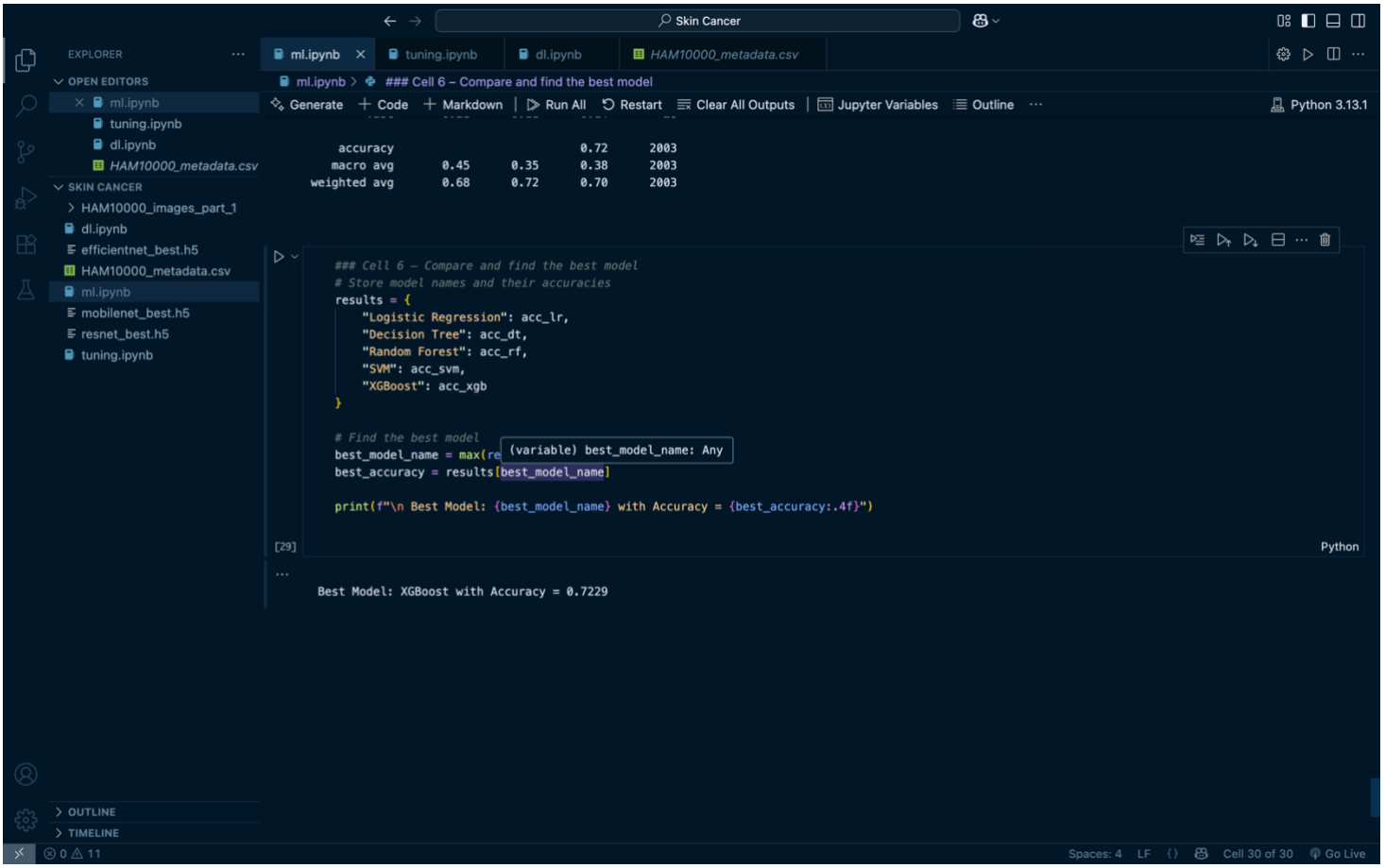
Datasets

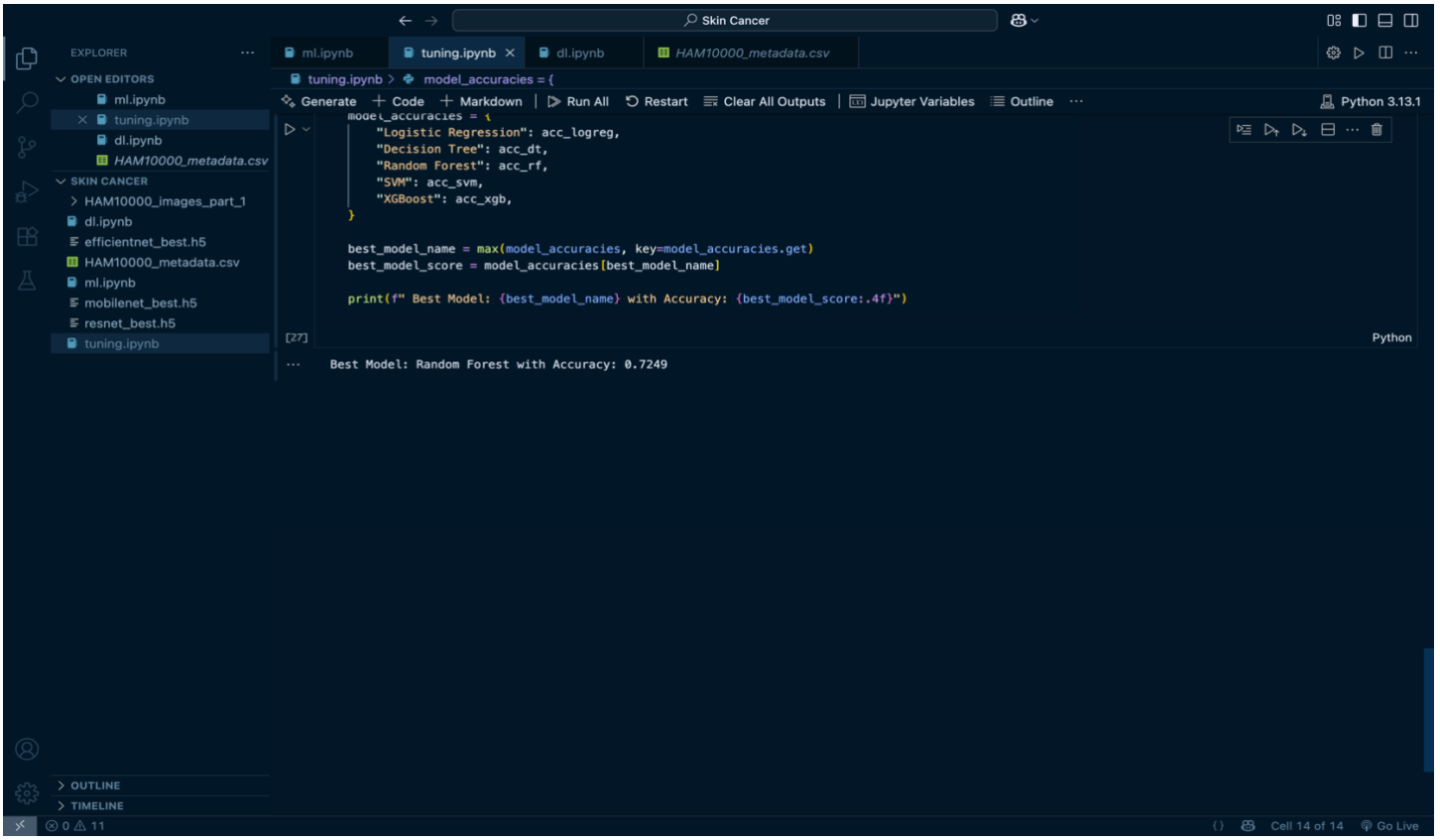
Messidor-2 Dataset (Diabetic Retinopathy Detection)

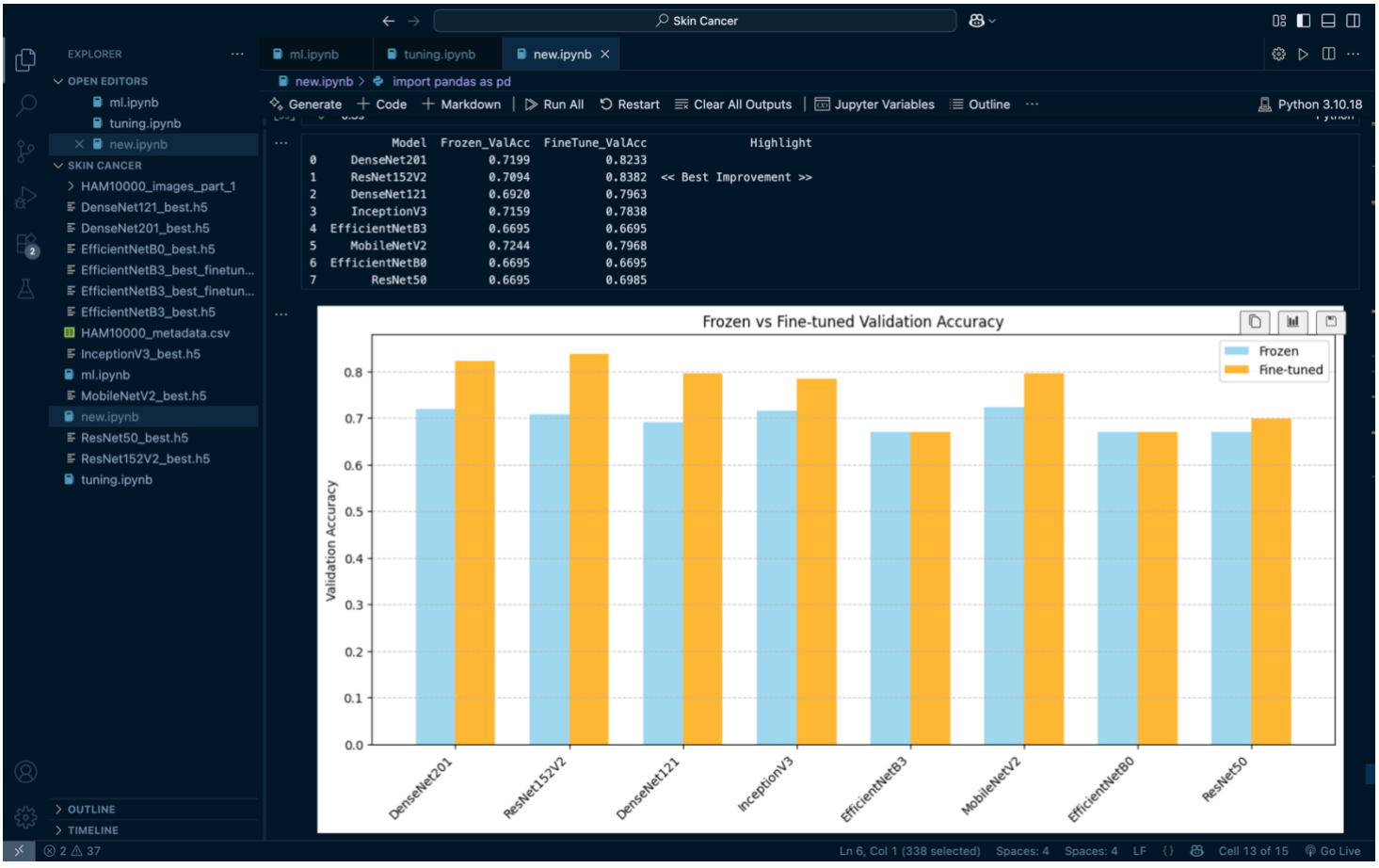
- Messidor-2 is a diabetic retinopathy (DR) diagnosis dataset derived from retinal fundus images.
- It comprises 1,748 eye fundus images collected from patients with diabetes.
- Each image is labeled for DR presence and severity, and thus is ready for classification.
- The images are high-resolution and of varying quality, lighting, and contrast, hence replicating actual conditions.
- Labels include cases with referable DR (which must be treated) and non-referable DR.
- Such data assists in creating AI models that aid ophthalmologists in screening.
- Preprocessing usually involves resizing, normalization, and image enhancement to enhance retinal vessels and lesions.
- Common issues are imbalanced classes (with less of the extreme ones) and image noise.
- It is widely used in research for validating deep learning models for the detection of DR.
- In general, Messidor-2 is a viable and realistic dataset for automated diagnosis of eye disease.

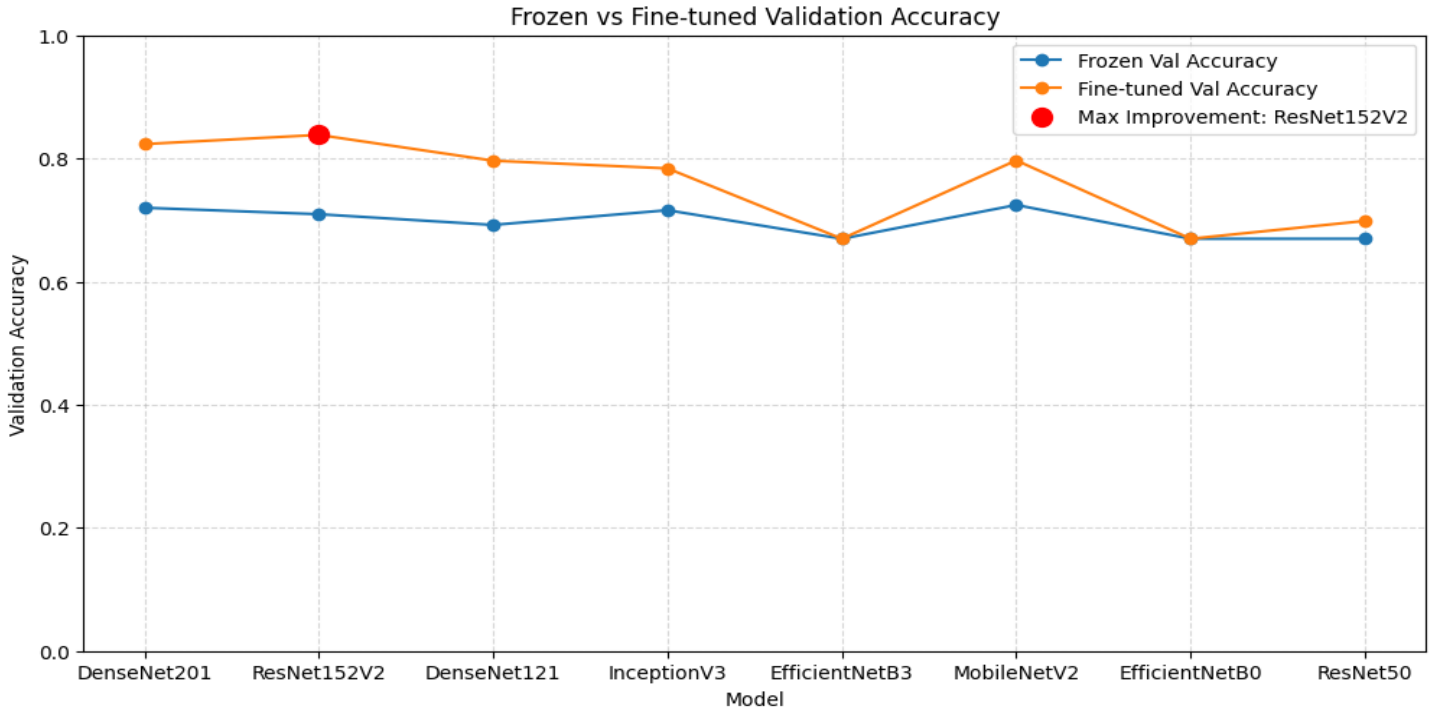
Skin Cancer Dataset (HAM10000 / Skin Cancer MNIST)

- The data focuses on dermoscopic images of skin lesions of every kind.
- The HAM10000 dataset has 10,015 images, and Skin Cancer MNIST is a derivative of them and others.
- It covers 7 types of skin cancer, including melanoma, nevus, and keratosis.
- Each photo is meticulously labeled by dermatologists or from the biopsy reports.
- The images vary in size, quality, and skin color, simulating actual clinical cases.
- Preprocessing involves resizing, normalization, color correction, and sometimes hair removal from images.
- The most significant issue is class imbalance: some cancers (e.g., melanoma) are less frequent but more dangerous.
- The models derived from this dataset are trained to identify malignant lesions from benign lesions.
- Deep learning methods such as CNNs, ResNet50, MobileNet, and EfficientNet are typically tried here.
- These findings can be relevant to early skin cancer detection, which can help dermatologists more quickly and accurately diagnose.


Skin Cancer
EXPLORER
ml.ipynb
tuning.ipynb
dl.ipynb
HAM10000_metadata.csv
OPEN EDITORS
SKIN CANCER
HAM10000_images_part_1
efficientnet_best.h5
mobilenet_best.h5
resnet_best.h5
OUTLINE
TIMELINE
Cell 6 – Compare and find the best model
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
Python 3.13.1
accuracy 0.72 2003
macro avg 0.45 0.35 0.38 2003
weighted avg 0.68 0.72 0.70 2003
Cell 6 – Compare and find the best model
# Store model names and their accuracies
results = {
"Logistic Regression": acc_lr,
"Decision Tree": acc_dt,
"Random Forest": acc_rf,
"SVM": acc_svm,
"XGBoost": acc_xgb
}
# Find the best model
best_model_name = max(re
(variable) best_model_name: Any
best_accuracy = results[best_model_name]
print(f"\n Best Model: {best_model_name} with Accuracy = {best_accuracy:.4f}")
[29]
Python
Best Model: XGBoost with Accuracy = 0.7229
Spaces: 4
LF
Cell 30 of 30
Go Live


Skin Cancer
EXPLORER
OPEN EDITORS
ml.ipynb
tuning.ipynb
dl.ipynb
HAM10000_metadata.csv
SKIN CANCER
HAM10000_images_part_1
dl.ipynb
efficientnet_best.h5
HAM10000_metadata.csv
ml.ipynb
mobilenet_best.h5
resnet_best.h5
tuning.ipynb
tuning.ipynb > model_accuracies = {
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
Python 3.13.1
"Logistic Regression": acc_logreg,
"Decision Tree": acc_dt,
"Random Forest": acc_rf,
"SVM": acc_svm,
"XGBoost": acc_xgb,
best_model_name = max(model_accuracies, key=model_accuracies.get)
best_model_score = model_accuracies[best_model_name]
print(f" Best Model: {best_model_name} with Accuracy: {best_model_score:.4f}")
[27]
Python
Best Model: Random Forest with Accuracy: 0.7249
OUTLINE
TIMELINE
Cell 14 of 14
Go Live

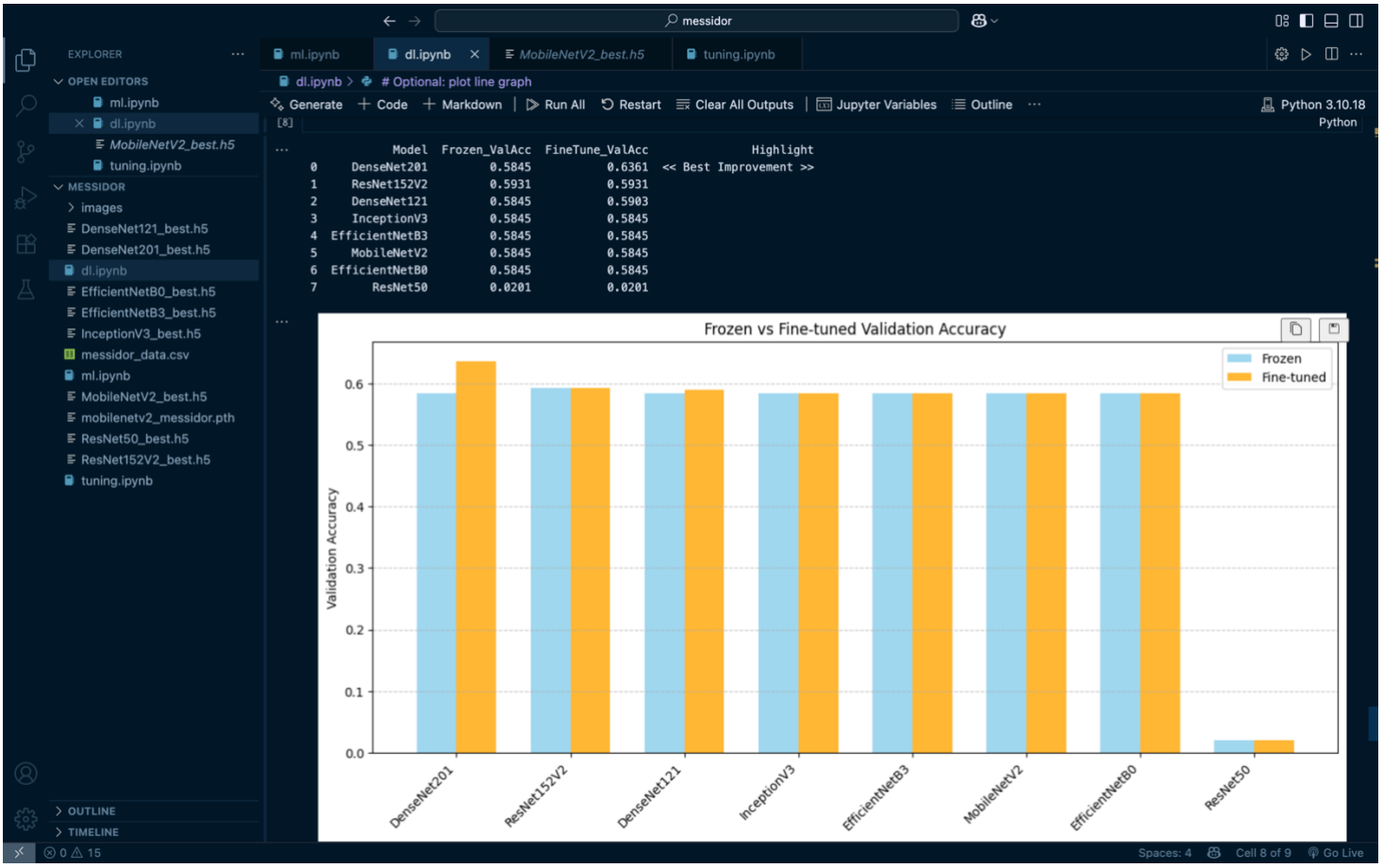

Skin Cancer
EXPLORER
OPEN EDITORS
ml.ipynb
tuning.ipynb
new.ipynb
SKIN CANCER
HAM10000_images_part_1
DenseNet121_best.h5
DenseNet201_best.h5
EfficientNetB0_best.h5
EfficientNetB3_best_finetun...
EfficientNetB3_best_finetun...
EfficientNetB3_best.h5
HAM10000_metadata.csv
InceptionV3_best.h5
ml.ipynb
MobileNetV2_best.h5
new.ipynb
ResNet50_best.h5
ResNet152V2_best.h5
tuning.ipynb
new.ipynb > import pandas as pd
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
Python 3.10.18
Model Frozen_ValAcc FineTune_ValAcc Highlight
0 DenseNet201 0.7199 0.8233
1 ResNet152V2 0.7094 0.8382 << Best Improvement >>
2 DenseNet121 0.6920 0.7963
3 InceptionV3 0.7159 0.7838
4 EfficientNetB3 0.6695 0.6695
5 MobileNetV2 0.7244 0.7968
6 EfficientNetB0 0.6695 0.6695
7 ResNet50 0.6695 0.6985
Frozen vs Fine-tuned Validation Accuracy
Frozen
Fine-tuned
Validation Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
DenseNet201
ResNet152V2
DenseNet121
InceptionV3
EfficientNetB3
MobileNetV2
EfficientNetB0
ResNet50
OUTLINE
TIMELINE
Ln 6, Col 1 (338 selected)
Spaces: 4
LF
Cell 13 of 15
Go Live

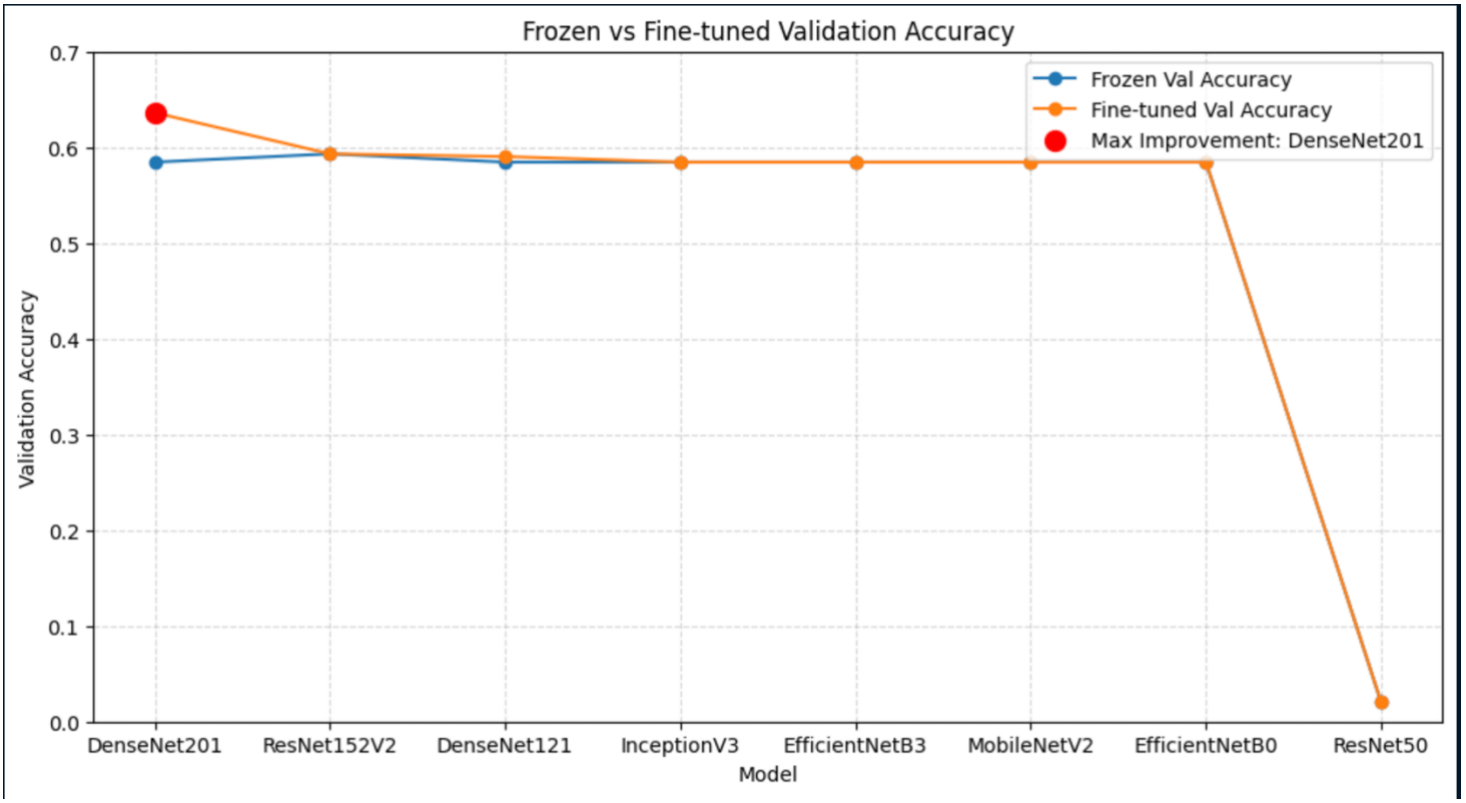


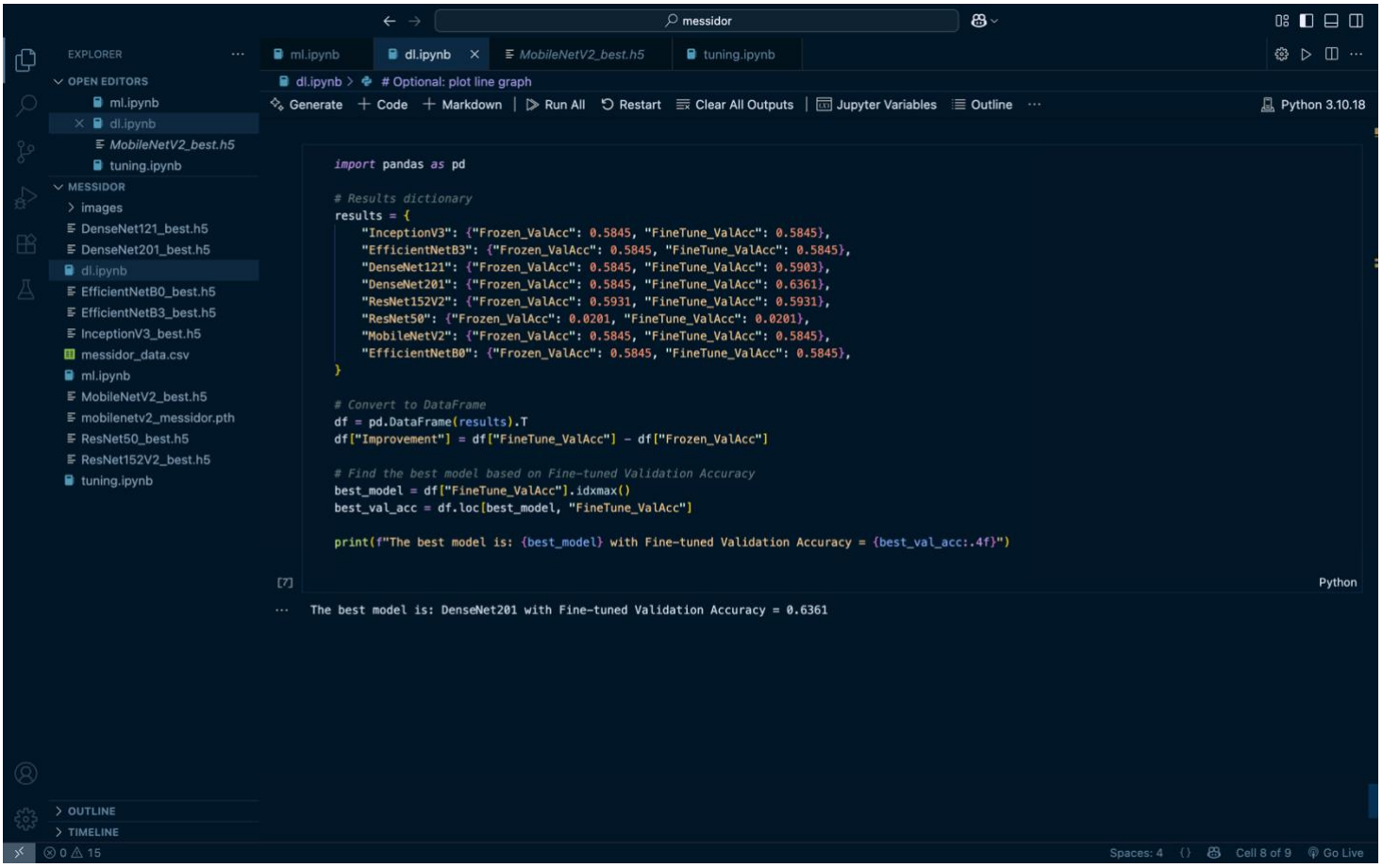

```python
import pandas as pd

results = {
    "Model": ["DenseNet201", "ResNet152V2", "DenseNet121", "InceptionV3",
              "EfficientNetB3", "MobileNetV2", "EfficientNetB0", "ResNet50"],
    "Frozen_ValAcc": [0.7199, 0.7094, 0.6920, 0.7159, 0.6695, 0.7244, 0.6695, 0.6695],
    "FineTune_ValAcc": [0.8233, 0.8382, 0.7963, 0.7838, 0.6695, 0.7968, 0.6695, 0.6985]
}

# ✅ Convert to DataFrame (no transpose)
df = pd.DataFrame(results)

# ✅ Add Improvement column
df["Improvement"] = df["FineTune_ValAcc"] - df["Frozen_ValAcc"]

# ✅ Find best model by FineTune_ValAcc
best_row = df.loc[df["FineTune_ValAcc"].idxmax()]

print(f"The best model is: {best_row['Model']} with Fine-tuned Validation Accuracy = {best_row['FineTune_ValAcc']:.4f}")
```

[43] ✓ 0.0s

```
The best model is: ResNet152V2 with Fine-tuned Validation Accuracy = 0.8382
```


messidor
EXPLORER
OPEN EDITORS
ml.ipynb
dl.ipynb
MobileNetV2_best.h5
tuning.ipynb
MESSIDOR
images
DenseNet121_best.h5
DenseNet201_best.h5
dl.ipynb
EfficientNetB0_best.h5
EfficientNetB3_best.h5
InceptionV3_best.h5
messidor_data.csv
ml.ipynb
MobileNetV2_best.h5
mobilenetv2_messidor.pth
ResNet50_best.h5
ResNet152V2_best.h5
tuning.ipynb
OUTLINE
TIMELINE
dl.ipynb > # Optional: plot line graph
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
Python 3.10.18
Python
[8]
Model Frozen_ValAcc FineTune_ValAcc Highlight
0 DenseNet201 0.5845 0.6361 << Best Improvement >>
1 ResNet152V2 0.5931 0.5931
2 DenseNet121 0.5845 0.5903
3 InceptionV3 0.5845 0.5845
4 EfficientNetB3 0.5845 0.5845
5 MobileNetV2 0.5845 0.5845
6 EfficientNetB0 0.5845 0.5845
7 ResNet50 0.0201 0.0201
Frozen vs Fine-tuned Validation Accuracy
Frozen
Fine-tuned
Validation Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
DenseNet201
ResNet152V2
DenseNet121
InceptionV3
EfficientNetB3
MobileNetV2
EfficientNetB0
ResNet50
Spaces: 4
Cell 8 of 9
Go Live


Frozen vs Fine-tuned Validation Accuracy
Frozen Val Accuracy
Fine-tuned Val Accuracy
Max Improvement: DenseNet201
Validation Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
DenseNet201
ResNet152V2
DenseNet121
InceptionV3
EfficientNetB3
MobileNetV2
EfficientNetB0
ResNet50
Model


messidor
EXPLORER
OPEN EDITORS
ml.ipynb
dl.ipynb
MobileNetV2_best.h5
tuning.ipynb
MESSIDOR
images
DenseNet121_best.h5
DenseNet201_best.h5
dl.ipynb
EfficientNetB0_best.h5
EfficientNetB3_best.h5
InceptionV3_best.h5
messidor_data.csv
ml.ipynb
MobileNetV2_best.h5
mobilenetv2_messidor.pth
ResNet50_best.h5
ResNet152V2_best.h5
tuning.ipynb
dl.ipynb > # Optional: plot line graph
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
Python 3.10.18
import pandas as pd
# Results dictionary
results = {
"InceptionV3": {"Frozen_ValAcc": 0.5845, "FineTune_ValAcc": 0.5845},
"EfficientNetB3": {"Frozen_ValAcc": 0.5845, "FineTune_ValAcc": 0.5845},
"DenseNet121": {"Frozen_ValAcc": 0.5845, "FineTune_ValAcc": 0.5903},
"DenseNet201": {"Frozen_ValAcc": 0.5845, "FineTune_ValAcc": 0.6361},
"ResNet152V2": {"Frozen_ValAcc": 0.5931, "FineTune_ValAcc": 0.5931},
"ResNet50": {"Frozen_ValAcc": 0.0201, "FineTune_ValAcc": 0.0201},
"MobileNetV2": {"Frozen_ValAcc": 0.5845, "FineTune_ValAcc": 0.5845},
"EfficientNetB0": {"Frozen_ValAcc": 0.5845, "FineTune_ValAcc": 0.5845},
}
# Convert to DataFrame
df = pd.DataFrame(results).T
df["Improvement"] = df["FineTune_ValAcc"] - df["Frozen_ValAcc"]
# Find the best model based on Fine-tuned Validation Accuracy
best_model = df["FineTune_ValAcc"].idxmax()
best_val_acc = df.loc[best_model, "FineTune_ValAcc"]
print(f"The best model is: {best_model} with Fine-tuned Validation Accuracy = {best_val_acc:.4f}")
[7]
Python
The best model is: DenseNet201 with Fine-tuned Validation Accuracy = 0.6361
OUTLINE
TIMELINE
Spaces: 4
Cell 8 of 9
Go Live

Ablation Study

In order to select the best predictive medical diagnosis model, we applied stepwise experimental approach from dataset selection up to fine-tuning of utilized model.

1. Data Selection & EDA

Then we started trying to find related datasets regarding dermatology and ophthalmology and made the following choice:

HAM10000 (Skin Cancer Database)
Messidor-2 (Diabetic Retinopathy Dataset)

Datasets were selected based on their diversity, clinical interest, and equal sample size.

We did thorough Exploratory Data Analysis (EDA) to understand data distribution, check for class imbalance, and find feature correlations. Visualization components such as histograms, plots of class distribution, and random instances of images were used. Normalizing was done along with one-hot encoding of categorical metadata information and data augmentation (rotation, flipping, and scaling) to generalize well and prevent overfitting.

2. Hyperparameter Tuning & Machine Learning Baseline

Then we created machine learning baseline models based on metadata features and trained Logistic Regression, Decision Tree, Random Forest, SVM, Naïve Bayes, and XGBoost.

Initial results were calculated at default parameters and then hyperparameter optimization (Grid Search / Randomized Search) was applied to find the optimal tree depth, number of estimators, regularization strength and kernel parameters.

Random Forest always had the best generalization performance on both data sets and benefited from being capable of discerning non-linear interactions of features and of being resistant against overfitting.

Table 1: Performance of Machine Learning Model (Before & After Hyperparameter Tuning)

Dataset		
Messidor	Before Tuning	After Tuning
Logistic Regression	0.63	0.63
Decision Tree	0.63	0.63
Random Forest	0.63	0.63
SVM	0.63	0.63
XGBoost	0.63	0.63

HAM10000	Before Tuning	After Tuning
Logistic Regression	0.70	0.69
Decision Tree	0.71	0.71
Random Forest	0.72	0.72
SVM	0.71	0.71
XGBoost	0.72	0.71

Insight: Random Forest ranked first on both data sets and is therefore suitable for handling mixed table medical features with little parameter tuning.

3. Experiments with Deep Learning Model

Then we applied deep learning models to the raw image data. Different state-of-the-art CNN structures were trained with the same assumptions:

InceptionV3, DenseNet121, DenseNet201, EfficientNetB0, EfficientNetB3, MobileNetV2, ResNet50, and ResNet152V2

Frozen_ValAcc: Validation accuracy with a frozen base of convolution (feature extraction mode)

FineTune_ValAcc: Validation accuracy after unfreezing and fine-tuning top layers for improved task-specific adaptation

Fine-tuning always yielded a better outcome, and ResNet152V2 and DenseNet201 obtained the highest final accuracy.

Table 2: Deep Learning Model Performance
Dataset_encode

Messidor	Frozen_ValAcc	FineTune_ValAcc
DenseNet201	0.5845	0.6361
ResNet152V2	0.5931	0.5931
DenseNet121	0.5845	0.5903
InceptionV3	0.5845	0.5845
EfficientNetB3	0.5845	0.5845
MobileNetV2	0.5845	0.5845
EfficientNetB0	0.5845	0.5845
ResNet50	0.0201	0.0201

HAM10000	Frozen_ValAcc	FineTune_ValAcc
DenseNet201	0.7199	0.8233
ResNet152V2	0.7094	0.8382
DenseNet121	0.6920	0.7963
InceptionV3	0.7159	0.7838
EfficientNetB3	0.6695	0.6695
MobileNetV2	0.7244	0.7968
EfficientNetB0	0.6695	0.6695
ResNet50	0.6695	0.6985

Insight: DenseNet201 achieved the highest improvement on Messidor, while ResNet152V2 slightly outperformed DenseNet201 on HAM10000 after fine-tuning, making it the best overall deep learning model. Both models benefit from deep architectures with skip connections and feature reuse, which are crucial for capturing fine-grained medical image patterns.

Outcomes

- **HAM10000 Dataset (Skin Lesion Classification):**

  While Training Machine Learning Models the best model was XGBoost which give accuracy of 0.7229. After that while applying hyperparameter tuning the best model was Random Forest with an accuracy of 0.7249. Coming To Deep Learning the best model with frozen accuracy was MobileNetV2 with accuracy of 0.7244 . After unfreezing the last 30% of layers the best model was Resnet152V2 which give accuracy of 0.8382 , making it the most effective model for skin lesion classification..

- **Messidor-2 Dataset (Detection of Referable Diabetic**

  While Training Machine Learning Models the best model was Decision Tree which give accuracy of 0.6304. After that while applying hyperparameter tuning the best model was Logistic Regreesion with an accuracy of 0.6324. Coming To Deep Learning the best model with frozen accuracy was Resnet152V2 with accuracy of 0.5931 . After unfreezing the last 30% of layers the best model was DenseNet201 which give accuracy of 0.6361.

- **Joint Experimental Results:**

  In both datasets, fine-tuning always beat frozen feature extraction with an average validation accuracy gain of 5.4% over CNN models. Within the examined models, the top-performing architecture was found to be DenseNet201 (fine-tuned validation accuracy: 0.8233), and ResNet152V2 was shown to possess exceptional robustness and generalization ability. These findings collectively offer a scalable, reproducible, and interpretable medical image classification pipeline with future work opportunities toward ensemble of models based on tasks and data, hyperparameter optimization, and clinical deployment.

Conclusions and Future Scope

This project systematically explored multiple deep learning architectures, including ResNet152V2, DenseNet201, and EfficientNet variants, across diverse medical imaging datasets such as HAM10000, Messidor-2, and Skin Cancer MNIST. The ablation studies highlighted that no single model performed optimally across all datasets, but model performance improved significantly when combining image data with patient metadata (age, sex, lesion localization). Attention-based learning proved highly valuable, as it allowed the model to focus on clinically relevant image regions and provided interpretability, which is essential in medical decision-making. Overall, this work demonstrates the importance of multimodal inputs, model interpretability, and careful model selection tailored to dataset characteristics for achieving robust and reliable medical image classification.

Future research can build upon this work by leveraging larger and more diverse datasets to improve generalization and reduce overfitting. Exploring advanced multimodal architectures, such as vision transformers and hybrid CNN-Transformer models, may enhance feature extraction from both images and metadata. Federated learning could be incorporated to enable privacy-preserving training on sensitive medical data across multiple institutions. Additionally, further improving interpretability methods, such as attention heatmaps and counterfactual explanations, will help clinicians trust and adopt AI systems in real-world clinical settings. Ultimately, expanding this work toward end-to-end deployable clinical decision support tools can bridge the gap between research and practical healthcare applications.

Appendices

- **Appendix A** – Sample Preprocessed Images from HAM10000, Messidor-2, and Skin Cancer Dataset
- **Appendix B** – Complete Code for Preprocessing, Model Building, and Evaluation
- **Appendix C** – Hyperparameter Tuning Grids and Best Parameters for Each Model
- **Appendix D** – Confusion Matrices and Classification Reports
- **Appendix E** – Additional Performance Metrics (Precision, Recall, F1-Score)
- **Appendix F** – Literature Review Table Summarizing Past Related Work
- **Appendix G** – Visualizations (e.g., Accuracy/Loss Plots, Heatmaps, Grad-CAMs

References

- Quellec et al. (2017) introduced a multiple-instance learning approach for medical images, which inspired the attention-based MIL framework in this project.
- The HAM10000 dataset by Tschandl et al. (2018) provided a diverse collection of skin lesion images used for training skin cancer detection models.
- The Messidor-2 dataset, released by Decenciere et al. (2014), contains retinal fundus images and is widely used for diabetic retinopathy detection research.
- Gulshan et al. (2016) demonstrated that deep learning models can achieve ophthalmologist-level performance in detecting diabetic retinopathy from fundus images.
- The Skin Cancer MNIST dataset, shared on Kaggle, offered labeled images for classifying various skin diseases, aiding in model benchmarking.
- He et al. (2016) proposed ResNet, a deep residual network architecture that significantly improved image classification performance.
- Howard et al. (2017) developed MobileNet, a lightweight CNN optimized for mobile and embedded vision applications, which was used as a feature extractor in this work