CS640 Project Report

Chaitanya Chakka, Astha Rastogi, Gagan Singal, Satya Akhil Galla

1. Introduction

The ISIC 2024 Challenge presents a unique opportunity to advance the field of dermatology by leveraging cutting-edge machine learning techniques to tackle the critical task of skin lesion analysis. As skin cancer remains one of the most prevalent and potentially fatal forms of cancer worldwide, early and accurate diagnosis is paramount. This competition is hosted by the International Skin Imaging Collaboration (ISIC) and aims to foster innovation in automated image analysis to aid clinicians in the diagnosis and management of skin lesions.

This report outlines the solution developed for the ISIC 2024 Challenge. The focus of the solution is to address the specific task of classifying and segmenting skin lesion images using advanced deep learning techniques. The approach combines state-of-the-art neural network architectures, robust data preprocessing strategies, and thoughtful post-processing steps to maximize performance.

Key aspects of this solution include:

- Model Architecture: The selection and fine-tuning of powerful deep learning models tailored for medical imaging tasks.
- Data Augmentation and Preprocessing: Strategies to enhance generalization by handling dataset imbalances and improving image quality.
- Evaluation and Metrics: Implementation of robust evaluation criteria to ensure reliable performance on unseen data.

This document delves into the technical details of the solution, including the methodologies adopted, experimental results, and an analysis of performance. By addressing the challenges posed by this competition, this work contributes to the broader effort of improving automated tools for dermatological diagnosis, with the goal of enhancing patient outcomes.

2. Data Analysis

The ISIC dataset consists of a total of 320,847 data points covering information about the lesions in terms of color, size, location, etc. These data points have been extracted by using the 3D Total Body Photographs (TBP) and the dataset consists of meta data as well as the images related to them. A keen observation common to both modals is the extreme imbalance present in the classes – samples pertaining to benign lesions far outweigh (320,533 samples) the malignant ones (314 samples). Each model has been processed with different techniques for the uni-modal models which are briefed in this section.

2.1. Tabular Data

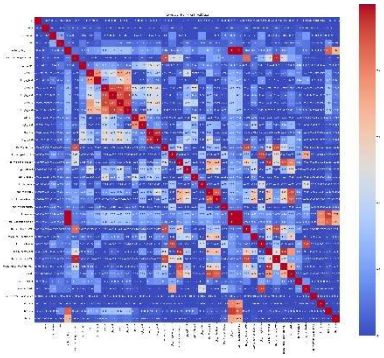
During the analysis, we identified missing values in the fields 'age-approx', 'sex', and 'anatom-site-general'. After careful consideration, we decided to remove the corresponding data points to maintain data integrity and avoid introducing biases through imputation. This led to removal of 15,314 samples out of which 9 are positive samples while the rest are negative samples.

To evaluate feature importance, we generated a correlation matrix to observe the relationships between

features and their associations with the target variable. While there was no feature that had a significant correlation with the target variable, we did notice a strong correlation between the fields 'anatom-site-general', 'tbp-lv-location', and 'tbp-lv-location-simple', as observed in the correlation matrix in Fig 1. This high correlation is consistent with the fact that these features represent increasingly specific categorizations of the same anatomical information.

As the correlation matrix did not give us a clear indicator of which features could be removed, we applied for Random Forest Classifier using Gini entropy as the criterion for feature selection. We opted for this method due to its effectiveness in identifying the most informative features while handling nonlinear interactions and complex relationships in the data. After analyzing the results shown in Fig 2, we decided to remove the 5 features with the least important score, as the remaining were quite similar in their importance. These features are 'sex', 'tbp-tile-type', 'anatom-site-general', 'tbp-lv- location-simple', 'tbp-lv-location'. From the results of the correlation matrix, we can see that these were also the most correlated features. Therefore, doing this mitigated redundancy while also simplifying the model.

To tackle the imbalance in the dataset, we have employed a combination of under sampling and oversampling along with class weights. During hyper parameter tuning, we randomly sampled 20%-30% of the dataset and rebalanced the dataset by using Synthetic Minority Oversampling Technique with Edited Nearest Neighbors (SMOTE-EEN) which is a variation of the standard SMOTE algorithm that oversamples the minority class and under samples the majority class by using the ENN algorithm [1]. This works by identifying the K nearest neighbors and then removing the datapoint that differs in label from the majority of the labels in the neighbors. We only rebalanced the dataset up to 30-70 ratio and then managed the rest of the balancing via modified loss function using class weights.



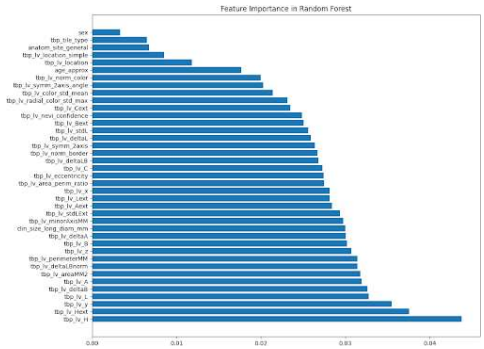


Figure 1: Correlation Matrix of all features
Figure 2: Random Forest Feature Importance

2.2. Image Data

A thorough understanding of the provided image dataset and labels is critical for developing a robust solution for the ISIC 2024 Challenge. This section outlines the data analysis conducted to gain insights into the dataset, identify potential challenges, and guide the preprocessing and model development stages.

We combined the images with their labels from the tabular data to work on the image model. The final dataset consisted of labeled images, where each image corresponds to a specific type of skin lesion or condition. The labels represent the ground truth categories for classification tasks. The data analysis revealed the following insights:

Class Distribution: Positive Samples: 320,533 Negative Samples: 314
Imbalance Ratio(Approximate): 1020:1
Image Format: .jpeg
Original Dimensions: Varied from (41x41) to (259x259)
Resized Dimensions: 224x224
Color Space and Channels: 3 Channels – RGB Space

Henceforth, a data augmentation plan was devised to address challenges such as class imbalance and overfitting. Examples include:

- Oversampling underrepresented classes.
- Applying transformations (rotation, flipping, etc) while preserving the semantic meaning of the images.

Based on all these findings, we made the following inferences:

1. No significant missing or corrupt data was found, ensuring dataset integrity.
2. There was a huge imbalance in the data.
3. Need for under-sampling or over-sampling
4. The presence of so many images meant getting a pre-trained model and fine-tuning the model because of resource requirements.

Since we planned to use ImageNet pre-trained models we used normalization techniques with ImageNet parameters. The ISIC images were normalized on mean and standard deviation values of ImageNet for each color channel.

3. Methods

This section outlines the methods we utilized for uni-modal and multi-modal fronts.

3.1. Methods for Tabular Data

To train our model on tabular data, we explored models known to perform well for classification tasks:

- **SVM with Radial Basis Function Kernel**: It can capture complex relationships between features in the dataset, which makes it highly effective for non-linear classification tasks while also handling outliers robustly.
- **Logistic Regression**: It is computationally efficient and is a standard baseline for binary classification. It is also less prone to overfitting
- **XGBoost**: It leverages ensemble learning by using multiple decision trees to improve prediction accuracy, robustness, and reduce overfitting.

We trained these models on the tabular dataset after performing normalizations and one-hot encoding for the categorical features. To determine the optimal hyperparameters, we employed Bayesian Optimization using Tree Parzan Estimator (TPE) instead of the traditional Grid Search. This model tries to model the relation between hyperparameters and an objective function (in this case the AUC score) as a probability distribution. For TPE method, we divide the hyperparameter space into two different distributions and use kernel density fitting to decide which distribution each combination goes to. One distribution represents good performance parameters and vice versa. Each iteration, we use expected improvement on each distribution and determine if we are going to get good results or not. This leads to strategic choices of parameters and thereby quicker convergence. This tuning was performed using Stratified K

fold validation with K = 5.

3.2. Methods for Image Data

We started by pre-processing the images by randomly choosing from the following:

- Rotating
- Dimension changing
- Shearing
- Zooming
- Flipping
- Changing Brightness and Contrast
- Normalization

Next, we divided training images into train and validation sets as well as incorporated stratified 3-fold method into our experiments.

For our experiments, we explored several models with different techniques. Primarily, we chose pre-trained models as the backbone, added fully connected layers on top of it and trained it on ISIC dataset. The models we experimented with are:

- VGG-19
- Inception-ResNet-v2
- EfficientNet-V2-B0

The motivation to work on models pre-trained on *ImageNet* is that they contain the information on how to extract meaningful features from images already, and we want to build additional feature extraction mechanisms specific to our dataset.

In our final method, we used EfficientNet-V2-B1.

We modified the model training in the following way:

- Sampling
- Data Augmentation

We chose different metrics to evaluate the model and settled for:

1. F1 Score:
    a. Combines precision and recall
    b. Provides good strategy through good and bad scores for too many negative predictions to tune the model further.
    c. Provides a balanced assessment of model performance

2. ROC AUC:

    a. Measures the model's ability to distinguish between classes
    b. Works across different classification thresholds

3.3. Method for Fusion

For the fusion model, we addressed data imbalance by under sampling negative samples by 99% and augmenting image data through techniques such as rotation, flipping, brightness contrast adjustment, resizing, and normalization.

The preprocessed data was then used to train a CNN based on the EfficientNet-V2-B1 architecture. The Adam optimizer was used for its weight decay regularization and adaptive learning rate capabilities. The Cosine Annealing Learning Rate was chosen as the scheduler to adjust the learning rate dynamically. The model was trained with a batch size of 4 for 5 epochs and 5-folds for cross-validation using Stratified K-folds. After preprocessing, the ratio of positive to negative samples per fold is 10:1. The out-of-fold predictions from this model were then appended to the tabular data.

To expand the dataset, we used 34 features of the tabular data, along with 42 new engineered features such as *lesion_size_ratio, hue_contrast, 3d_volume_approximation*, etc using the following formulas:

- lesion_size_ratio = tbp_lv_minorAxisMM / clin_size_long_diam_mm
- hue_contrast = abs(tbp_lv_H - tbp_lv_Hext)
- volume_approximation_3d = tbp_lv_areaMM2 * sqrt(tbp_lv_x**2 + tbp_lv_y**2 + tbp_lv_z**2)

This gave us a total of 76 features that were fed to a Voting Classifier using soft voting. This combined XGBoost, LightGBM, and CatBoost models to leverage the strengths of each algorithm – namely XGBoost's robust predictive performance, LightGBM's high computational efficiency, and CatBoost's ability to process categorical features accurately.

4. Experiments and Results

4.1. Tabular

All the models have been hyperparameter tuned with the TPE Bayesian Optimization using the hyperopt library. The parameters utilized have been tabularized in Table 2.

Table 2: Best Hyperparameters and AUC scores.

Model	**Hyperparameter List**	**Best Training AUC**	**Best Validation AUC**
SVM	C=0.776, gamma='scale'	0.88	0.868
Logistic Regression	C=1.938, solver='newton-cholesky', max_iter=2000	0.91	0.896
XGBoost	Learning_rate=0.168, min_child_weight=100, max_depth=10, objective='binary:logistic', subsample=0.5, colsample_bytree=1/6, lambda=4.679	0.981	0.942

For our final outputs, we have considered the complete dataset. From our experiments we observed that among the models, XGBoost achieved the highest average AUC score across 5 folds of 0.93, indicating superior predictive performance compared to LR (0.89) and SVM (0.87).

The ROC curves for XGBoost and Logistic Regression provide further insights into the model's performance. The curve for XGBoost has a steeper rise towards the top left corner which reflectes a higher true positive rate at lower false positive rates compared to logistic regression. Therefore, XGBoost has a better ability to balance specificity and sensitivity compared to logistic regression. These findings show that XGBoost is more suitable for our task of tabular data classification.

Table 1: Tabular Data AUC Scores

Model	**AUC Score**
SVM	0.87
Logistic Regression	0.89
XGBoost	0.93

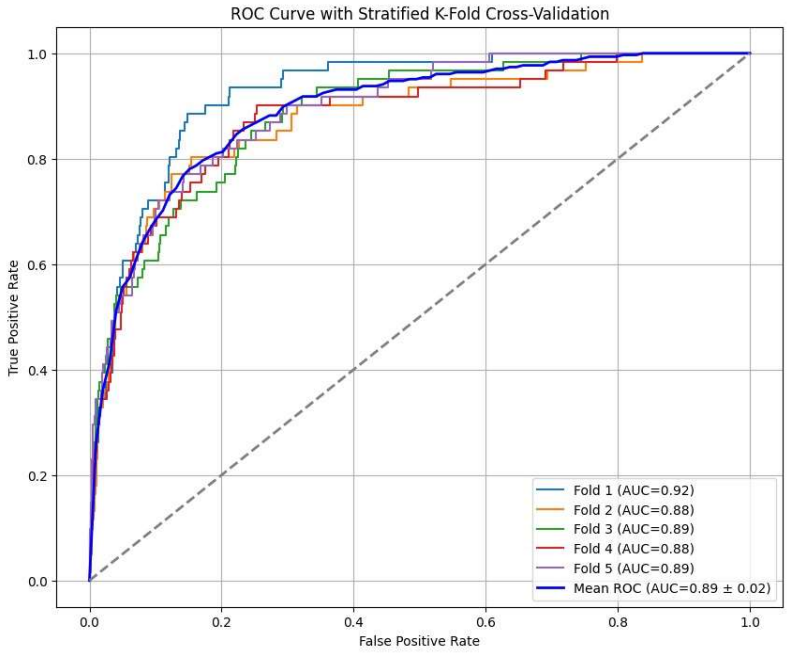


Fig: ROC Curve for Logistic Regression

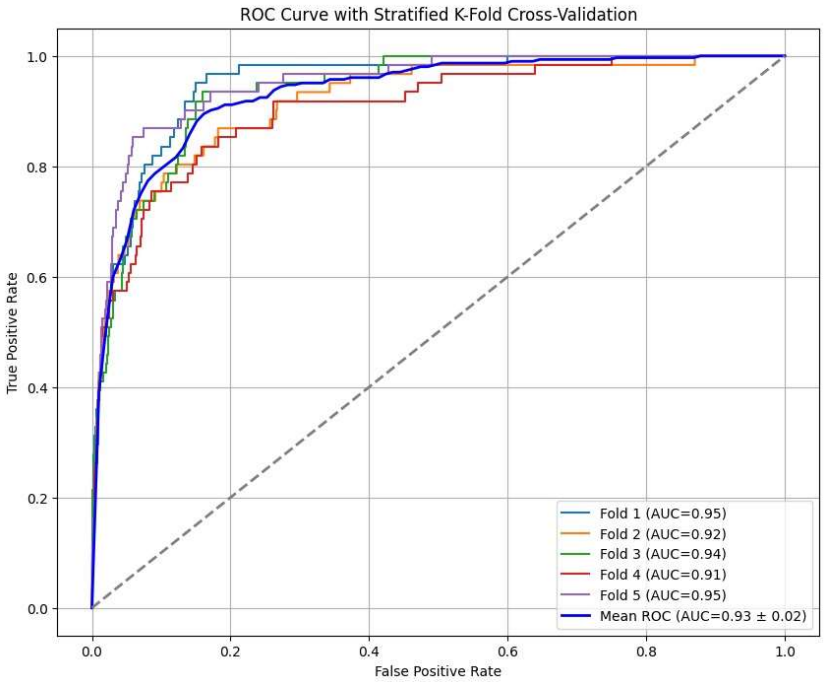


Fig: ROC Curve for XGBoost

4.2. Image

On top of the mentioned models in Section 3.2, we tried techniques like:

- *Image Augmentation*
- *Regularization*
- *Class weights*
- *Focal Loss*
- *Shifting Decision Boundary*
- *Sampling Techniques*.

During the experiments we made the following inferences:

1. Image Augmentation is necessary since the class imbalance is extremely high around 0.009%.
2. The idea behind Regularization is that while training the backbone model on ISIC data it is likely that the model could be overfit on negative classes and hence make use of L2 regularization to prevent it.
3. Similar ideas follow for class weights, where the weights are calculated based on the distribution of data and then passed on to loss function. The loss function penalizes the positive samples more than negative samples according to the class weights parameters resulting in better feature extraction of positive samples despite the low number.
4. We found that using the entire dataset was negatively impacting the training process while only using Image data. Therefore, to reduce the total datapoints, the number was chosen based on

experimentation. We tried *100k, 50k, 25k* datapoints and found that *300k* and *100k* did not have much difference in results except tending to overfit. *25k* datapoints were yielding much better metrics but it is a very small proportion of the dataset. Hence, we had to settle at *50k* datapoints as our optimal. But note that the total dataset has been used for the fusion model.

Model	Variant	Data Points	ROC/AUC	F1 Score
VGG-19 with augmented minority class	Base	50k, including all Minority Data	0.81	0.66
	L2 Norm	50k, including all Minority Data	0.77	0.63
	Class weights	50k, including all Minority Data	0.79	0.67
Inception-ResNet-v2 with augmented minority class	Base	100k, including all Minority Data	0.64	0.47
	L2 Norm + Class weights	100k, including all Minority Data	0.74	0.63
EfficientNet-v2-b1	High over-sampling of minority data	Whole dataset	0.96	0.79

Table 2: Performance of different models with various techniques and data configurations.

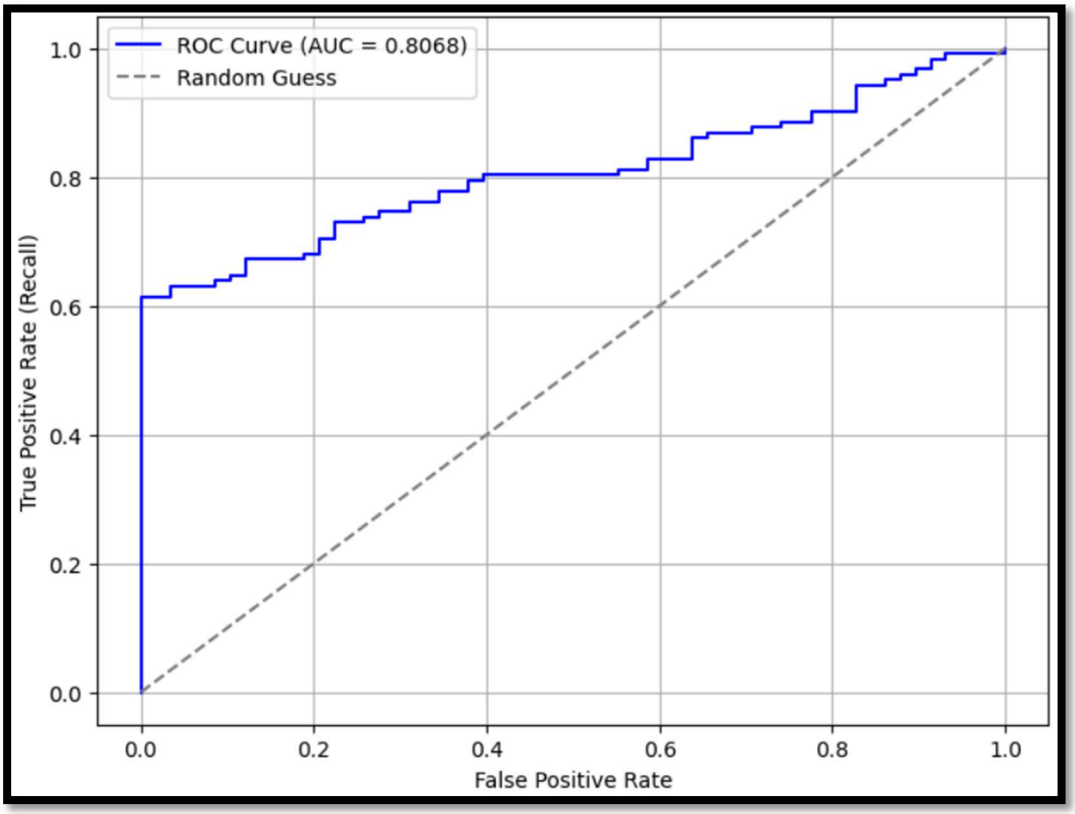


Fig: ROC Curve for VGG-19

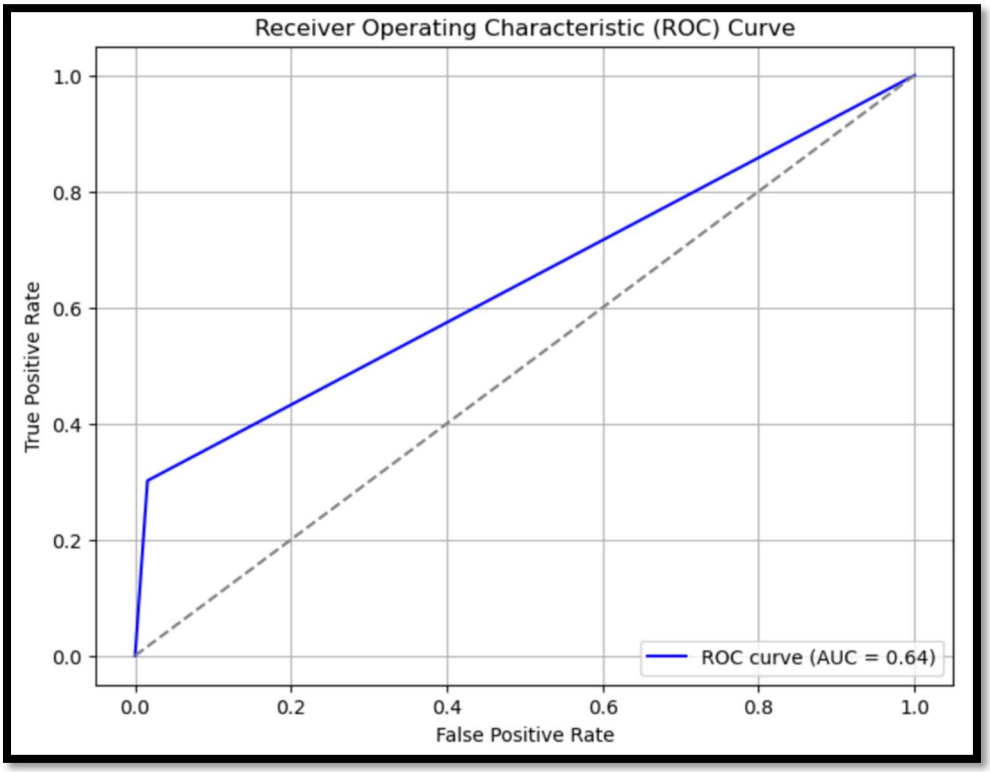


Fig: ROC Curve for Inception Model

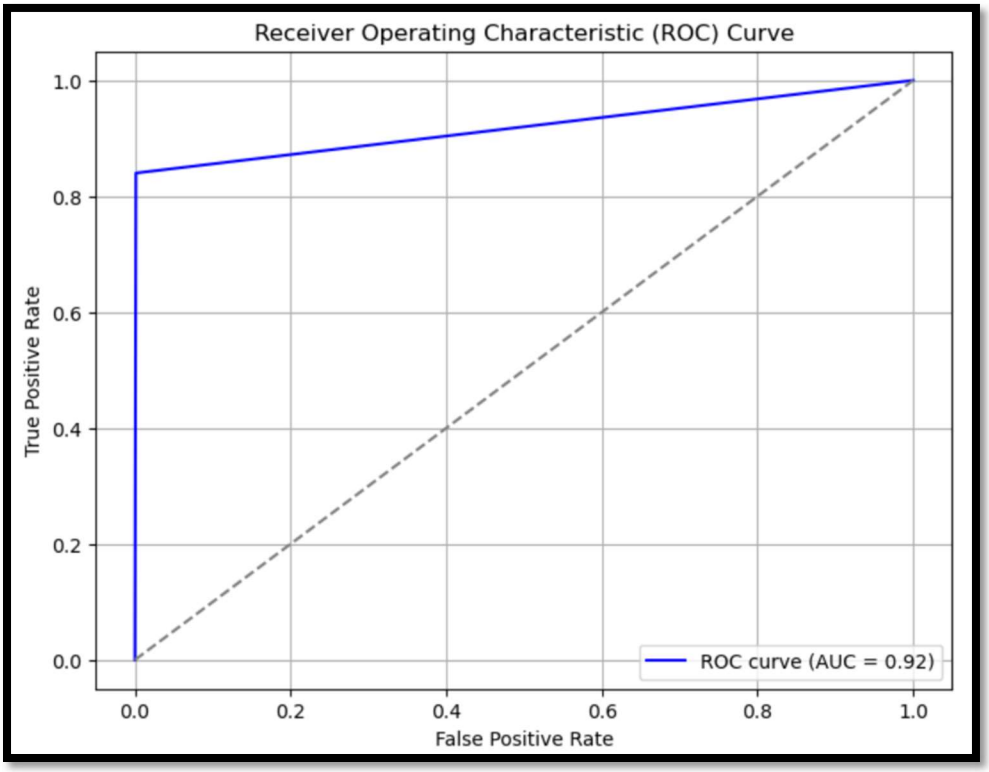


Fig: ROC Curve for EfficientNet Model

4.3. Fusion

The evaluation metric used for the fusion model was the AUC scores, along with the ROC curve. Using Stratified K-Fold Cross Validation across 5 folds, we got an average AUC of 0.96 on the validation set.

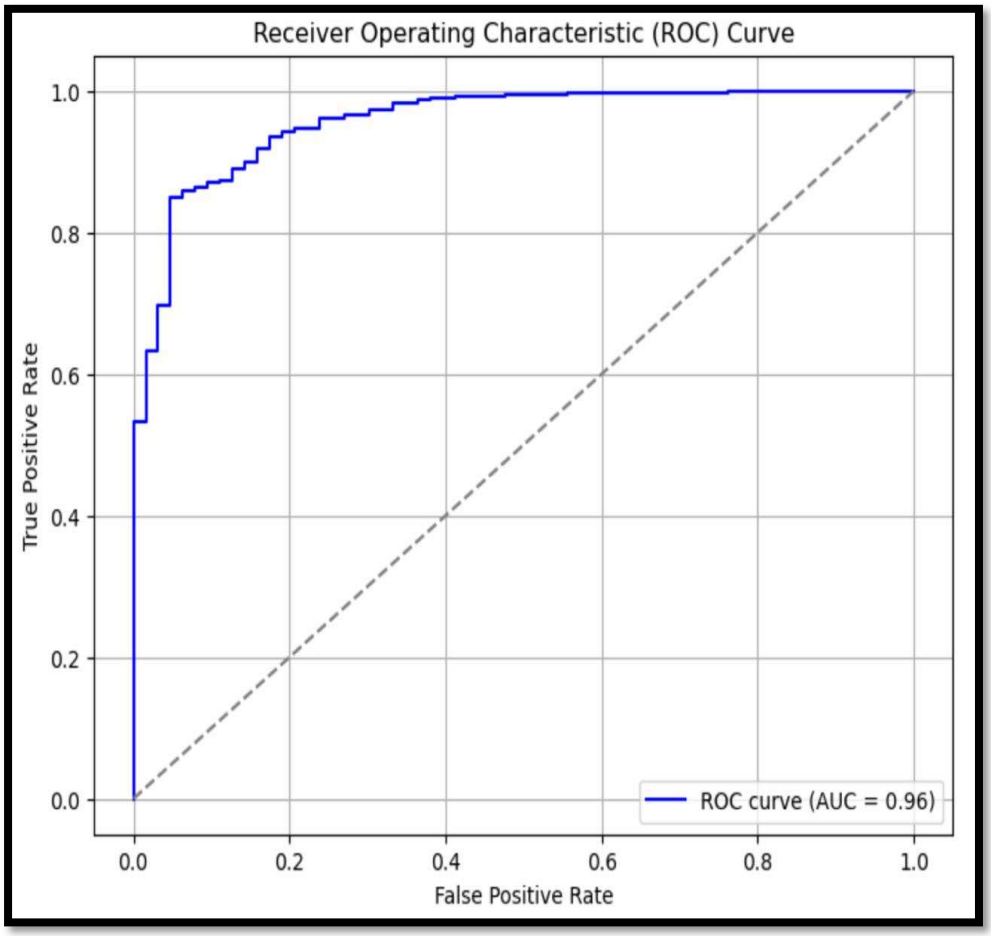


Fig: ROC Curve for Fusion Model

5. Discussions

5.1. Tabular Data

Observing the hyperparameter tuning results and the results, we see a slight variation in the AUC scores. The difference was that tuning utilized a maximum of 30% of the negative samples while the final experiments utilized the complete dataset. The same trend is observed in many of the leaderboard solutions where they utilized only 5%-10%.

The choice of removing only 5 features was made by testing a varied number of features and we determined that removing more than 5 was showing some noticeable change in performance. The advantage of using a percentage of whole dataset was also useful in turning into the model which led to quicker and larger search space for all the models. Nonetheless, SVM model taken significantly larger time to converge when compared to other models and at the end showed poor performance. The reason for longer times due to its cubic complexity and its extreme sensitivity to imbalanced data.

5.2. Image

Since we know that the data is biased it is reasonable to assume that positive samples have less probability of being classified correctly. The original decision boundary for a binary classifier is 0.5, therefore it is logical that shifting the boundary in favor of positive samples increases the confidence for malignant cells. The optimal value for the boundary is obtained from precision-recall curves. The precision and recall values were plotted across the threshold probabilities of 0 to 1. The optimal value was found when F1 was maximum signifying the balance of precision and recall.

In such a way, we have trained a model with augmented positive samples using class weights for further data balancing and L2 regularization. With the trained model we have obtained the optimal threshold value of 0.05. Upon testing the shifted model on test set we obtained an AUC score of 0.91, F1 score of 0.86 with precision of 0.78, recall of 0.95 and accuracy of 0.78. The metrics indicate better performance, but the confusion matrix contains high false negatives meaning that many of the malignant patients are being classified as benign. This is dangerous in practical application since our aim is to detect any possibility of cancer as early as possible with the help of deep learning models. Hence, we discarded this idea.

6. Conclusion and Future Work

In conclusion, despite the performance metrics with Image data the tabular data proved to be very useful and the tabular data could solely be sufficient, though the fusion model has slightly better performance. Using the images predictions in tabular data as an additional feature increased the metrics by a small margin. The best performance with tabular data was AUC 0.93 and the fusion model yielded an AUC of 0.96, whereas the best image model gave an AUC of 0.81. It denotes that tabular data has better information, moreover the tabular methods XGBoost, Logistic Regression, SVM have better ability to model the data despite the heavy data imbalance.

This shows that Deep Learning architectures are hard to use when the internal working of the model is unclear. We do not know how the model is trying to learn the patterns when the positive samples are very low. We had better performance with tabular methods because we used classic ML architecture and were able to tune the model according to observations with every experiment. We tried to do the same with CNNs but did not work out due to black box nature.

Hence choosing an appropriate model is important and tuning the model after every experimentation. For future works it could benefit that more of the features are extracted like mean value of each channel in RGB because they could correlate directly with positive or negative samples. Another potential approach could be to concatenate different cancer detection datasets to help with balancing.

7. Code Availability

/projectnb/cs640grp/students/chvskch/Fusion Model Code
The code is available at this path on SCC.

References

[1] Kumar, Vinod, et al. "Addressing binary classification over class imbalanced clinical datasets using computationally intelligent techniques." Healthcare. Vol. 10. No. 7. MDPI, 2022.
[2] https://www.kaggle.com/code/richolson/isic-2024-borrowed-179lb-tabular-of-imagenet