Evaluation documentation

1. Introduction

The evaluation setup is comprised of multiple parts:

- Getting the training data
- Preparing the training dataset
- Training the models on a k-fold split dataset
- Analyzing results (loss convergence, metrics, etc.)
- Collecting model .onnx files from each fold and each model trained
- Running inference script on the test set using an ensemble of models
- Analyzing results from inference

1.1 Evaluation Metrics

The metrics that are used for evaluating performance on training, validation and testing datasets are the following:

1. Accuracy
2. Precision
3. Recall (Sensitivity)
4. F1 Score
5. Specificity
6. Area Under the ROC Curve
7. Area Under the Precision-Recall Curve
8. Mean Loss
9. Confusion Matrix
10. Best Area Under the ROC Curve

1.1.1 Accuracy

Accuracy is the most intuitive metric—it calculates the proportion of correctly classified samples over the total number of samples. While it can give a quick snapshot of performance, it becomes less informative in imbalanced datasets. For example, if only 10% of the images are malignant and the model classifies all images as benign, it would still achieve 90% accuracy, despite failing entirely at detecting malignancies.

1.1.2 Precision

Precision measures how many of the samples predicted as positive (e.g., malignant) are actually positive. It's particularly important when the cost of false positives is high. In the context of melanoma detection, a false positive may cause unnecessary anxiety, further testing, or even biopsies. High precision means the model makes fewer of these costly mistakes.

1.1.3 Recall

Recall (Sensitivity), on the other hand, tells us how well the model identifies all the actual positives. A high recall means that most malignant cases are being detected. In medical applications, this is often more important than precision, because missing a melanoma case (false negative) could delay treatment and worsen outcomes.

1.1.4 F1 Score

F1 Score balances precision and recall by computing their harmonic mean. This is particularly useful when we care about both false positives and false negatives, and when the dataset is imbalanced. A high F1 score indicates that the model does well in both catching positive cases and avoiding false alarms.

1.1.5 Specificity

Specificity is the true negative rate—it tells us how well the model identifies benign cases. In other words, it's the counterpart to recall but for the negative class. It's important to ensure that benign lesions aren't mistakenly flagged as malignant, which could lead to overtreatment or anxiety.

1.1.6 AUROC

AUROC (Area Under the Receiver Operating Characteristic Curve) is a powerful metric that captures the trade-off between sensitivity (recall) and specificity across all classification thresholds. A model with AUROC = 1 is perfectly able to distinguish between the classes, while a score of 0.5 indicates random guessing. AUROC is particularly useful when the dataset is imbalanced, as it's less affected by class prevalence.

1.1.7 AUPRC

AUPRC (Area Under the Precision-Recall Curve) is often more informative than AUROC in the context of imbalanced data. It focuses on how well the model ranks true positives among all positive predictions and is especially helpful when we're primarily concerned with the minority class—like malignant cases in skin lesion datasets. This metric helps us evaluate how effectively the model isolates malignant samples from the bulk of benign ones.

1.1.8 Mean Loss

Mean Loss reflects how well the model is learning during training. It’s a direct measurement from the loss function (e.g., cross-entropy or focal loss), averaged per epoch. A decreasing training and validation loss over time usually signals better fitting to the training data. Tracking training and validation loss allows us to identify underfitting or overfitting trends early on.

1.1.9 Confusion Matrix

Confusion Matrix is a 2x2 table (in binary classification) that gives a detailed breakdown of predictions:

- True Positives (correctly identified malignant),
- False Positives (benign predicted as malignant),
- True Negatives (correctly identified benign),
- False Negatives (malignant predicted as benign).

This matrix gives us a concrete view of how and where the model is making mistakes, which can guide further improvements in data balancing, model architecture, or training procedure.

1.1.10 ROC AUC Best

ROC AUC Best is a checkpointing reference that keeps track of the highest AUROC score achieved on the validation set during training. This is especially useful for model selection—saving the model state that performs best in distinguishing between benign and malignant cases ensures we don’t deploy an underperforming model from an arbitrary epoch.

2. Results of model training

In this chapter, we will discuss the different results in the process of model training. We will highlight the different architecture's loss convergence, performance metrics etc. The different colors in the graphs represent the 5-Fold combinations.

2.1 ShuffleNetV2

2.1.1 Training and validation loss convergence

In the Figure below, training and validation loss convergence can be seen. The graphs demonstrate a complete alignment between the two loss functions, indicating a well-fitted model. After a certain number of epochs, for most folds, the validation loss stops decreasing, and early stopping is activated after 15 epochs of no change in validation loss.

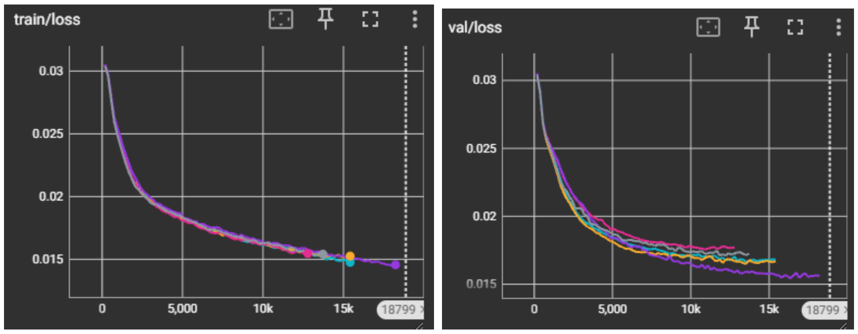

Figure 1 Training and validation loss for ShuffleNetV2

2.1.2 Validation metrics convergence through the epochs

To not clutter, we will focus on the most important metrics for this problem, which are Area Under the ROC Curve, Area Under the Precision-Recall Curve and F1-score. In the graphs below we can see that the metrics increase from epoch to epoch, meaning the model is learning the representations well.

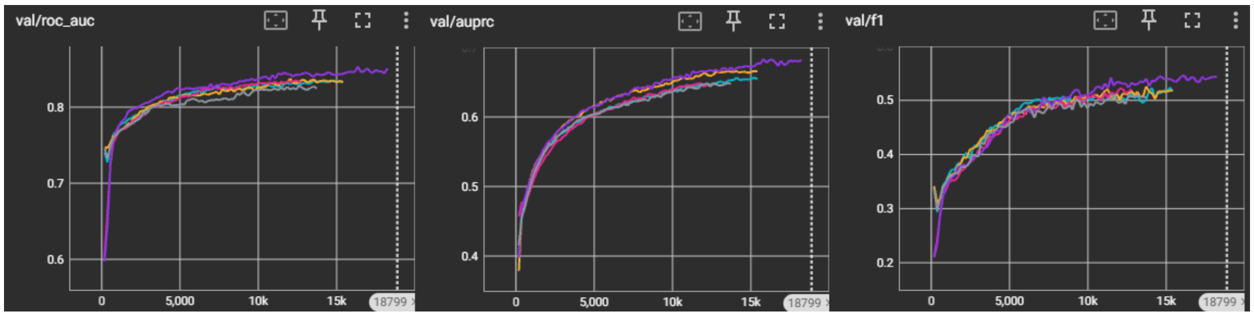

Figure 2 Validation AUROC, AUPRC, F1 through the epochs

2.1.3 Independent test set confusion matrices

In Figure 1, we can see the results of training the models on different folds. The results shown in the confusion matrices below are when the models are tested on an independent test set. The results show a consistency in the different aspects of the confusion matrices, meaning the model's hyperparameters are picked well and the model converges similarly, even when trained on different data.

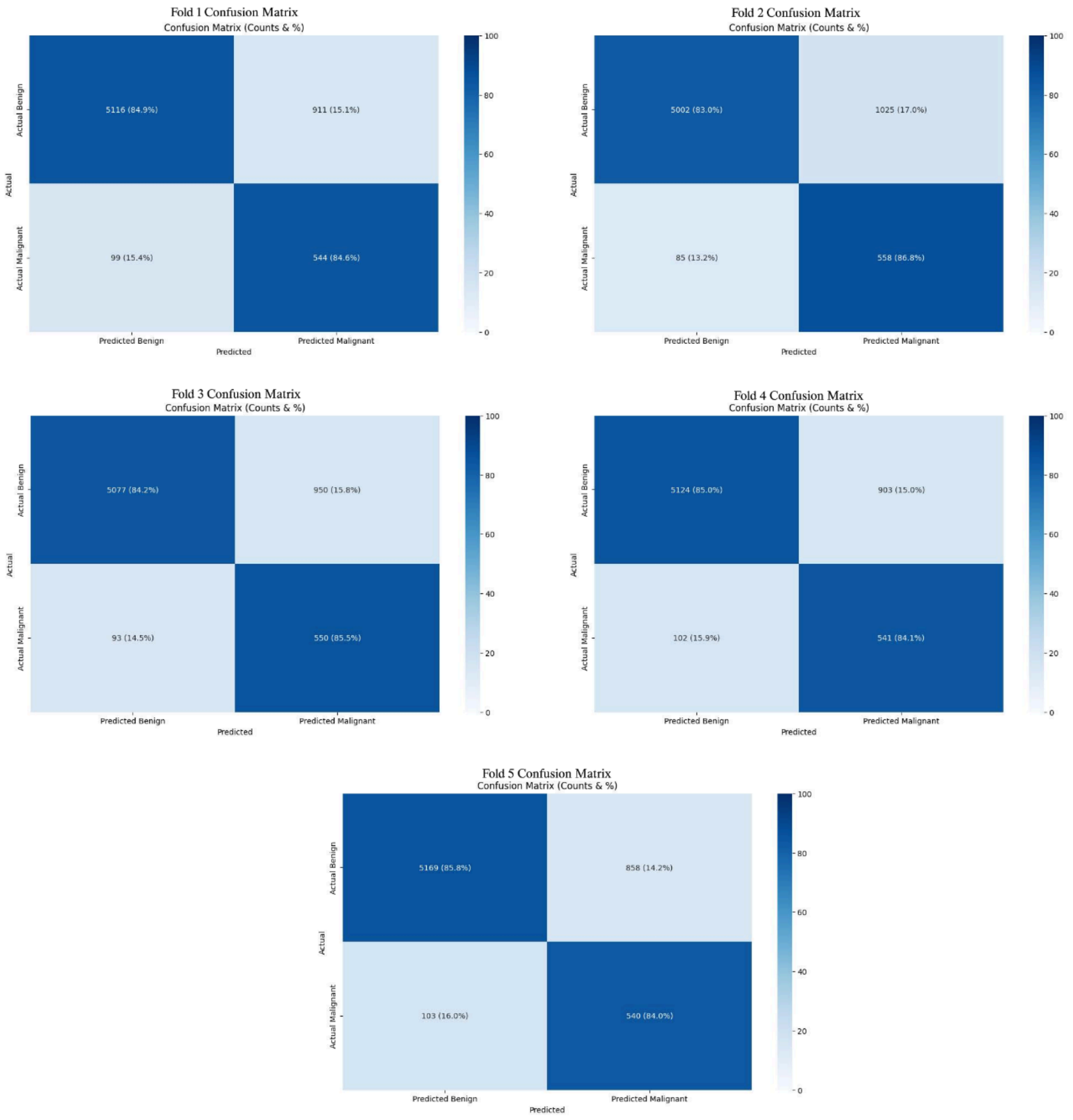

Figure 3 ShuffleNetV2 5-fold Confusion Matrices

2.1.4 Independent test set metrics

In the following table, the results of metrics for each combination of Fold training can be seen.

Table 1 Test set Metrics

Fold	Acc (%)	Spec (%)	Recall (%)	Prec (%)	F1 (%)	ROC AUC (%)	AUPRC (%)
1	84.86	84.88	84.60	37.39	51.86	84.74	65.21
2	83.36	82.99	86.78	35.25	50.13	84.89	67.21
3	84.36	84.24	85.54	36.67	51.33	84.89	65.95
4	84.93	85.02	84.14	37.47	51.84	84.58	67.47
5	85.59	85.76	83.98	38.63	52.92	84.87	69.14
Average	**84.62**	**84.58**	**85.01**	**37.48**	**51.82**	**84.79**	**67.40**

From the table it can be seen that across different folds, the testing performance is consistent. The precision is quite low, impacting the F1 score, but this is quite normal, as the dataset is unbalanced.

2.2 MobileNetV3

2.2.1 Training and validation loss convergence

In the figure below, training and validation loss convergence is shown.

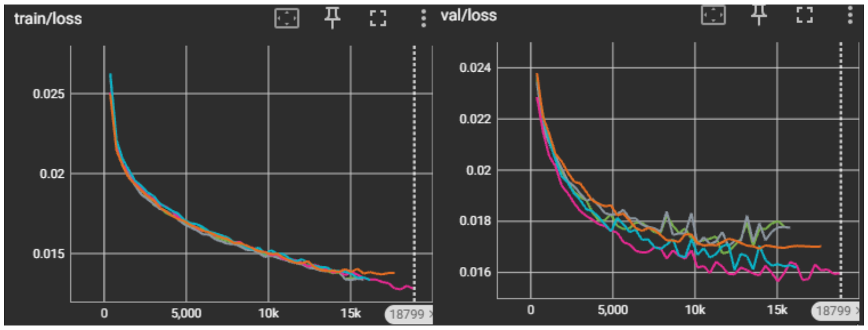

Figure 4 Training and Validation loss convergence

From the graphs it can be seen that the model's loss functions across the training and validation dataset converge well. However, in the later epochs the convergence on the validation set becomes a bit more unstable, meaning the model could be focusing on noise, leading to slight, inter-epoch overfitting. The trend of decreasing loss is still quite consistent.

2.2.2 Validation metrics convergence through the epochs

For the sake of brevity, we will only highlight the most important metrics, which are Area Under the ROC Curve, Area Under the Precision-Recall Curve and F1-score. In the graph below the convergence of these metrics can be seen, when looking through the entire training process.

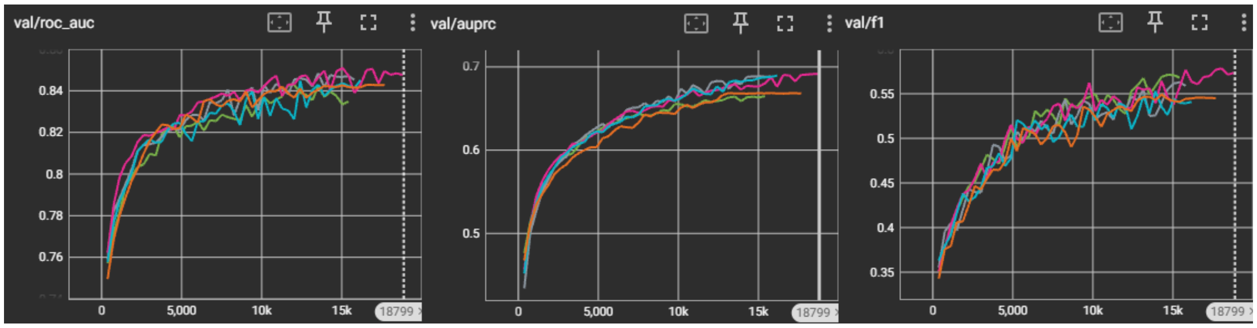

Figure 5 Validation AUROC, AUPRC, F1 through the epochs

The metrics converge well, meaning the model is learning the representations and how to differentiate between the malignant and benign class.

2.2.3 Independent test set confusion matrices

The results shown in the figure below represent the performance on different Fold combinations. The results show a consistency in TP, FP, FN, FP counts, meaning the model's hyperparameters are picked well and the model convergence similarly, even when trained on different data.

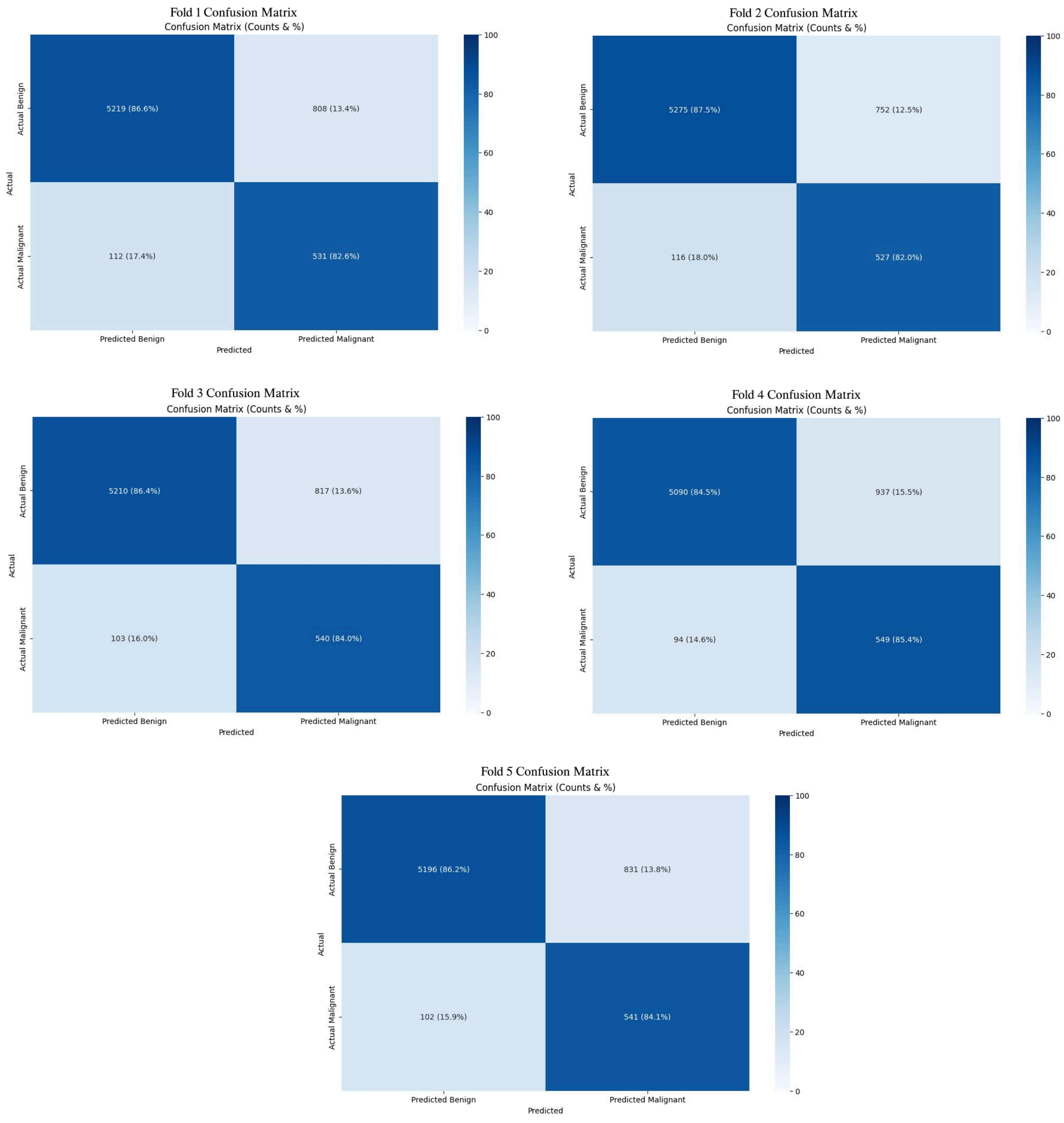

Figure 6 MobileNetV3 5-fold Confusion Matrices

2.2.4 Independent test set metrics

In the following table, the results of metrics for each combination of Fold training can be seen.

Table 2 Test set Metrics

Fold	**Acc (%)**	**Spec (%)**	**Recall (%)**	**Prec (%)**	**F1 (%)**	**ROC AUC (%)**	**AUPRC (%)**
1	86.21	86.59	82.58	39.66	53.58	84.59	70.74
2	86.99	87.52	81.96	41.20	54.84	84.74	69.82
3	86.21	86.44	83.98	39.79	54.00	85.21	71.08
4	84.54	84.45	85.38	36.94	51.57	84.92	70.51
5	86.01	86.21	84.14	39.43	53.70	85.17	70.46
Average	**85.99**	**86.24**	**83.61**	**39.40**	**53.14**	**84.93**	**70.12**

From the table it can be seen that across different folds, the testing performance is consistent. The precision is quite low, impacting the F1 score, but this is quite normal, as the dataset is unbalanced. The performance is slightly better than the ShuffleNetV2 model, when considering precision and AUPRC.

2.3 SqueezeNet1.1

2.3.1 Training and validation loss convergence

In the figure below, training and validation function for SqueezeNet1.1 can be seen. Apart from the Fold 1 (pink line), the models demonstrate a strong convergence.

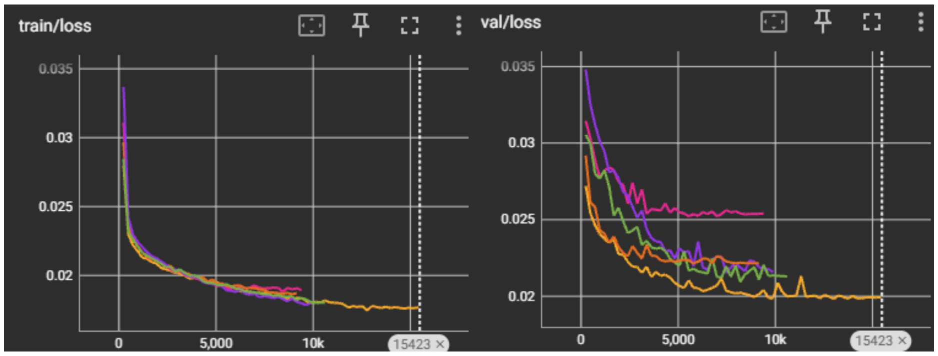

Figure 7 Training and validation loss through the epochs

2.3.2 Validation metrics convergence through the epochs

As with the previous models, in the following figure we will take a look at the AUROC, AUPRC and F1 score through the epochs to see if the most important performance metrics converge in a meaningful way.

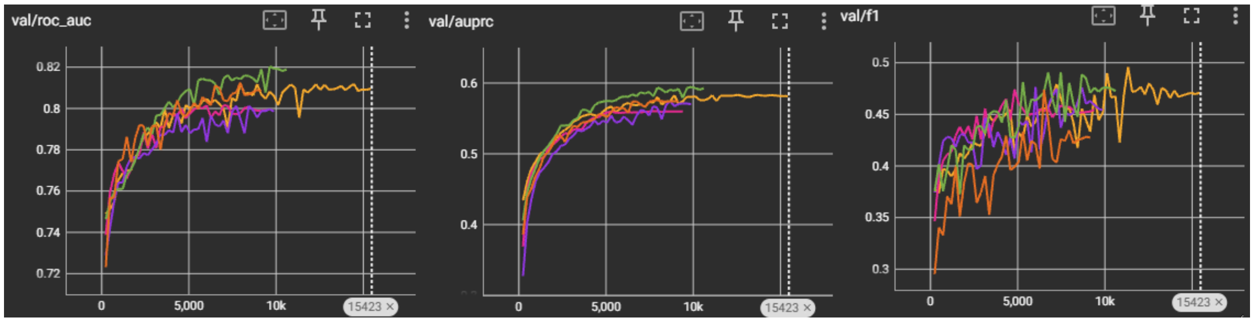

Figure 8 Validation AUROC, AUPRC, F1 through the epochs

From the graphs we can see that even though the AUROC and AUPRC scores converge nicely through the epochs, the F1 score convergence is somewhat unstable. Potentially, this could be because of the model's small size compared to the ShuffleNet and MobileNet architectures.

2.3.4 Independent test set confusion matrices

In the following figure, performance across different Folds can be seen in the form of results on the independent test sets, as shown in the confusion matrices.

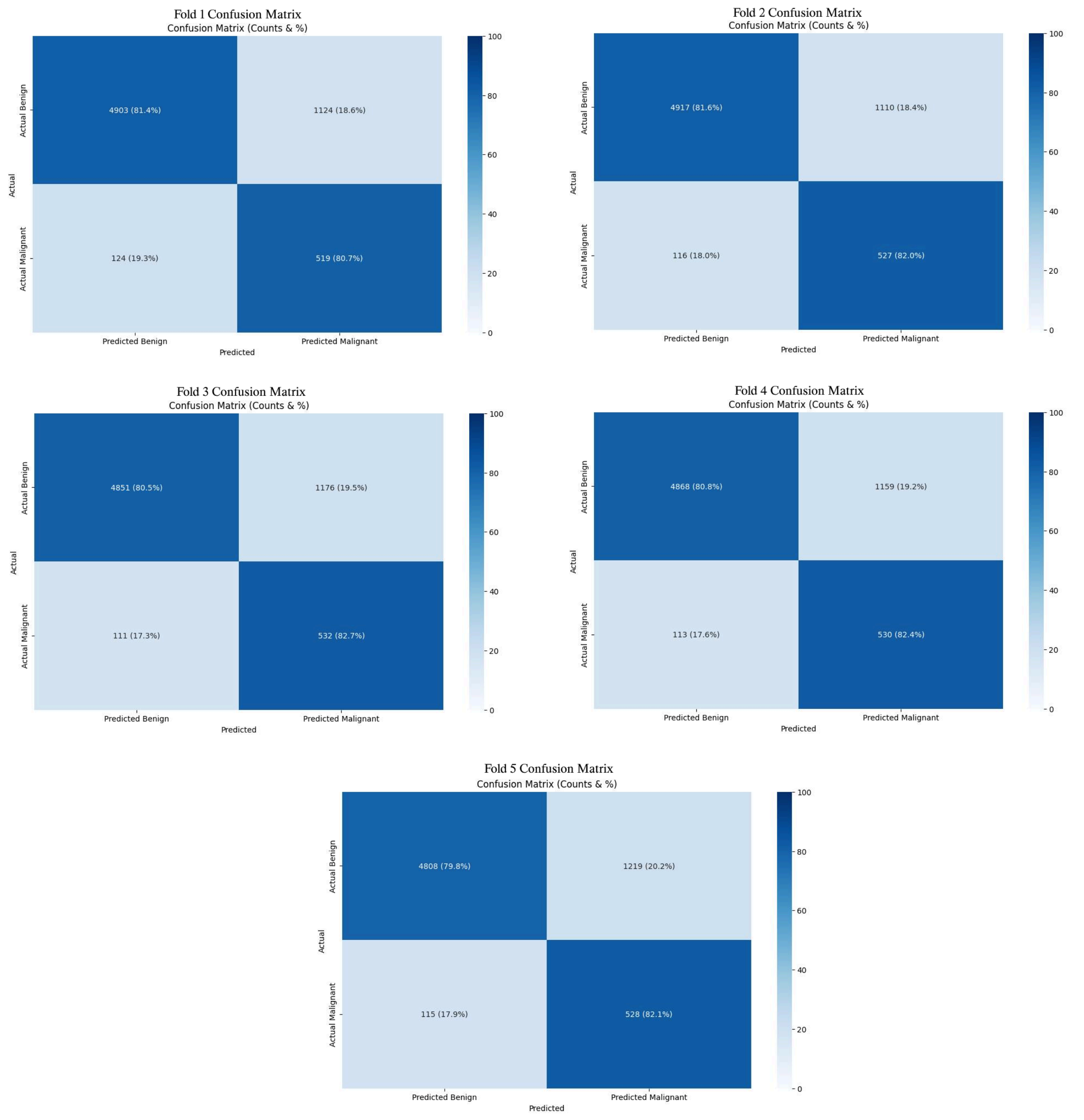

Figure 9 SqueezeNet1.1 5-fold Confusion Matrices

The performance across the board seems consistent, but Fold 1 is lagging behind a bit in terms of TP count.

2.3.4 Independent test set metrics

In the table shown below, we can see the test set metrics.

Table 3 Test set Metrics

Fold	**Acc (%)**	**Spec (%)**	**Recall (%)**	**Prec (%)**	**F1 (%)**	**ROC AUC (%)**	**AUPRC (%)**
1	81.29	81.35	80.72	31.59	45.41	81.03	57.07
2	81.62	81.58	81.96	32.19	46.23	81.77	59.75
3	80.70	80.49	82.74	31.15	45.26	81.61	57.52
4	80.93	80.77	82.43	31.38	45.45	81.60	59.57
5	80.00	79.77	82.12	30.22	44.18	80.94	58.30
Average	**80.91**	**80.79**	**81.99**	**31.31**	**45.31**	**81.39**	**58.44**

The table shows weaker performance in terms of Precision, but all metrics across the board don't seem to deviate much from one another, meaning the model's hyperparameters are good for this training, and it is fit well.

2.4 Model comparison and conclusions

The table shown below demonstrates the difference in performance between different architectures, in terms of test set metrics.

Table 4 Differences between architectures and performance indicators

Model	No. of params	Size (MB)	Average AUROC (%)	Average AUPRC (%)	Average F1 Score
ShuffleNetV2	1.3 million	5	84.79	67.40	51.82
MobileNetV3	3 million	11	84.93	70.12	53.14
SqueezeNet1.1	732k	2	81.39	58.44	45.31

What can be noticed is that SqueezeNet lags behind significantly, while ShuffleNet and MobileNet are somewhat similar, in terms of performance, despite the difference in the number of parameters and size. This might indicate that ShuffleNet is a better suiting architecture for this problem. We will still be using all of the trained models in the inferencing part, to see if the ensemble can even out the different shortcomings of each of the networks.

3. Results of inference using ensemble

In terms of performance and inference speed (while running on a GTX 1070 along with Ryzen 5 5600), each combination of the ensemble is highlighted in the table below, along with the performance metrics. These results represent the ensemble's performance on the test set, the same one on which the individual models were tested on. This is to see if there's any point in doing an ensemble in the first place.

Table 5 Different Ensembles independent test set performance results

Model Ensemble (number of models)	**Inference time per image (s)**	**Accuracy (%)**	**Specificity (%)**	**Recall (%)**	**Precision (%)**	**F1 (%)**	**AUROC (%)**	**AUPRC (%)**
MobileNetV3 (5)	0.017	86.72	86.94	84.60	40.87	55.12	93.71	72.47
ShuffleNetV2 (5)	0.017	84.86	84.85	84.91	37.42	51.95	93.19	68.64
SqueezeNet1.1 (5)	0.015	81.80	81.70	82.74	32.54	46.71	90.71	61.24
ShuffleSqueeze (10)	0.02	84.78	84.69	85.69	37.38	52.05	92.84	67.76
MobileSqueeze (10)	0.023	85.55	85.61	84.91	38.64	53.11	93.24	70.61
ShuffleMobile (10)	**0.025**	**86.46**	**86.59**	**85.23**	**40.41**	**54.83**	**94.02**	**73.03**
ShuffleMobileSqueeze (15)	0.03	85.92	86.01	85.07	39.35	53.81	93.64	71.59

The table demonstrates a significant improvement in terms of general performance (when compared to single model inference), meaning ensembling has been effective. Though it increases the processing time (not linearly, but still), it makes sense to do it.

The best combination of models seems to be the ShuffleMobile ensemble, when looking at general performance scores (AUROC, AUPRC, Recall), however it is slightly behind MobileNetV3 (5) in terms of F1 score. **For the final model,** we will use **the ShuffleMobile ensemble**, as it makes sense for it to be the best (more models, less variance in decision-making).

In the figures below we can see the ROC Curve and The Confusion Matrix for the best resulting ensembles result on the test set.

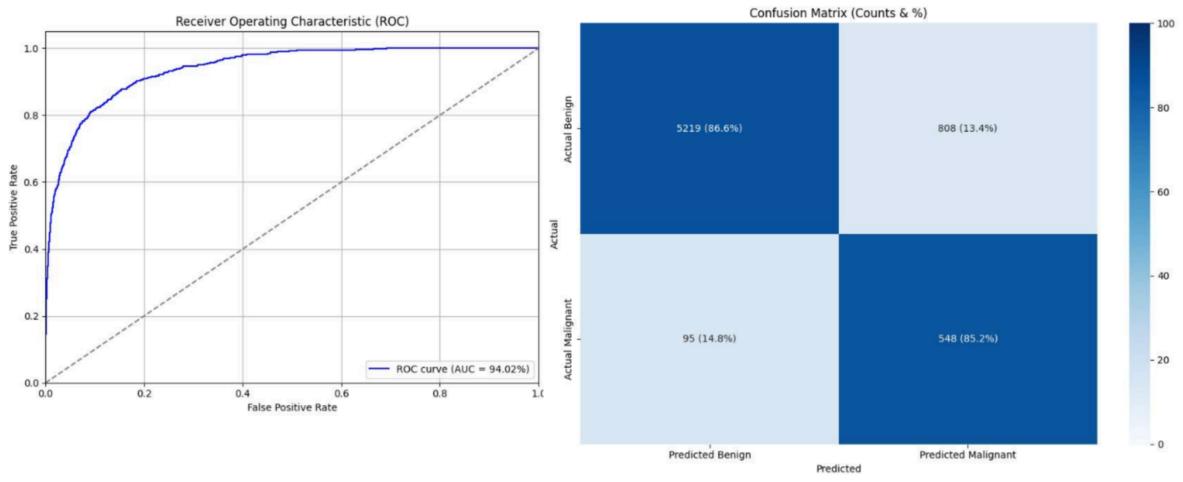

Figure 10 ROC Curve and Confusion Matrix for the ensemble test inferencing

The results demonstrate a well-balanced relation between Sensitivity and Specificity, as well as a good balance between Precision and Recall.

4. Conclusion

The models (SqueezeNet, MobileNet, and ShuffleNet) demonstrate strong performance, achieving a solid balance between precision and recall. The loss function shows consistent convergence across both training and validation datasets, suggesting that the models are learning effectively. Furthermore, the confusion matrix values remain stable across different fold combinations, reinforcing the robustness of the models.

The hyperparameter choices—such as batch sizes, learning rates and L2 regularization, and model selection—have proven to be effective, leading to consistent results. The ensemble methods have shown a noticeable improvement on the independent test set, validating the effectiveness of the technique. This enhancement suggests that the ensemble approach makes sense and contributes to a more robust generalization.

The pipeline overall strikes a good balance between model performance and inference speed. Given the nature of the models and the training setup, it is evident that the approach scales well in terms of accuracy, while maintaining reasonable inference times.

When **ensembling** the models, the **best performance** was achieved by the **Shuffle-Mobile** combination with an **AUROC of 94.02.** Because of that, SqueezeNet is excluded from the final ensemble model.

In conclusion, the overall pipeline is well-tuned, showing solid performance with a clear trade-off between model complexity and resource utilization, and is well-suited for large-scale inference tasks.