CAD Project: A SKIN LESION CLASSIFICATION APPROACH USING DEEP LEARNING

By

Xavier Beltran Urbano

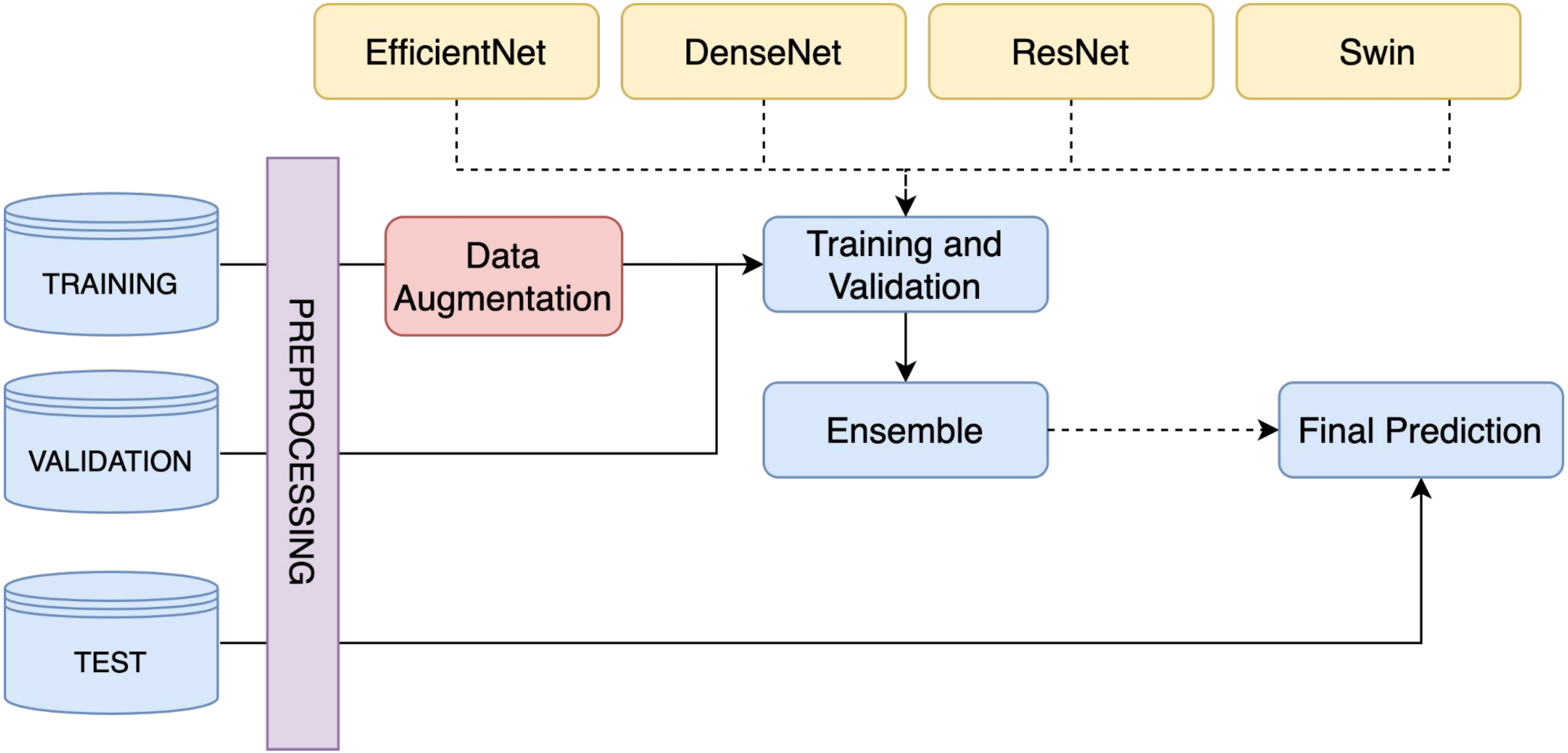
Muhammad Zain Amin

Table of Contents

01

INTRODUCTION

Main Pipeline



Skin Lesion Dataset

- The dataset contains high resolution and different sizes of images.

CAD Challenge 1: Binary Dataset

Types	Number of Images	
	Train	Validation
Nevus	7725	1931
Others	7470	1865
Total	15195	3796

Balanced dataset

CAD Challenge 2: Multiclass Dataset

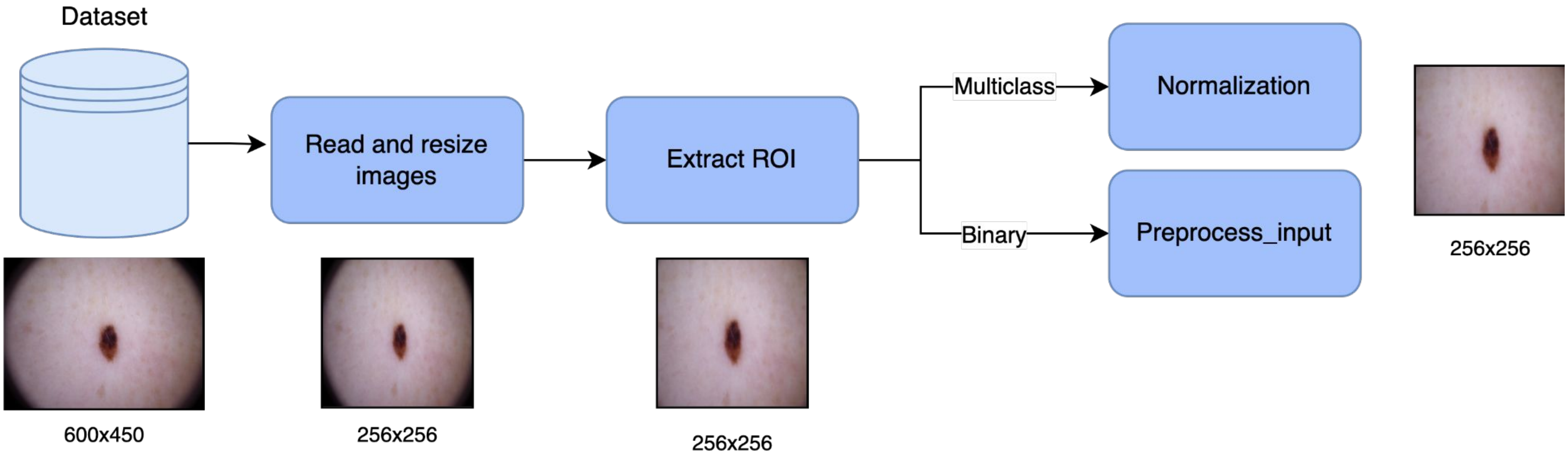
Lesion Types	Number of Images	
	Train	Validation
BCC	1993	498
Melanoma	2713	678
SCC	376	94
Total	**5082**	**1270**

Imbalance dataset

02

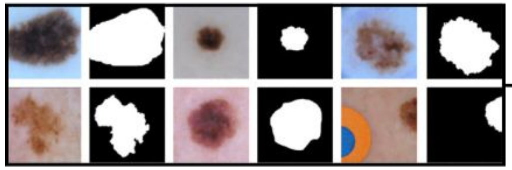
PREPROCESSING

General Preprocessing



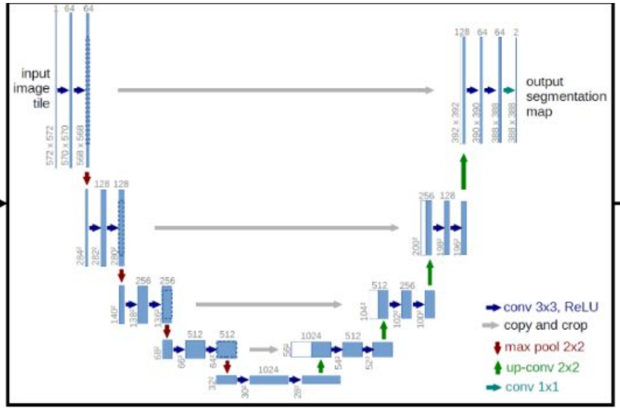
Segmentation Step

Segmentation step:



ISIC 2017

Original image

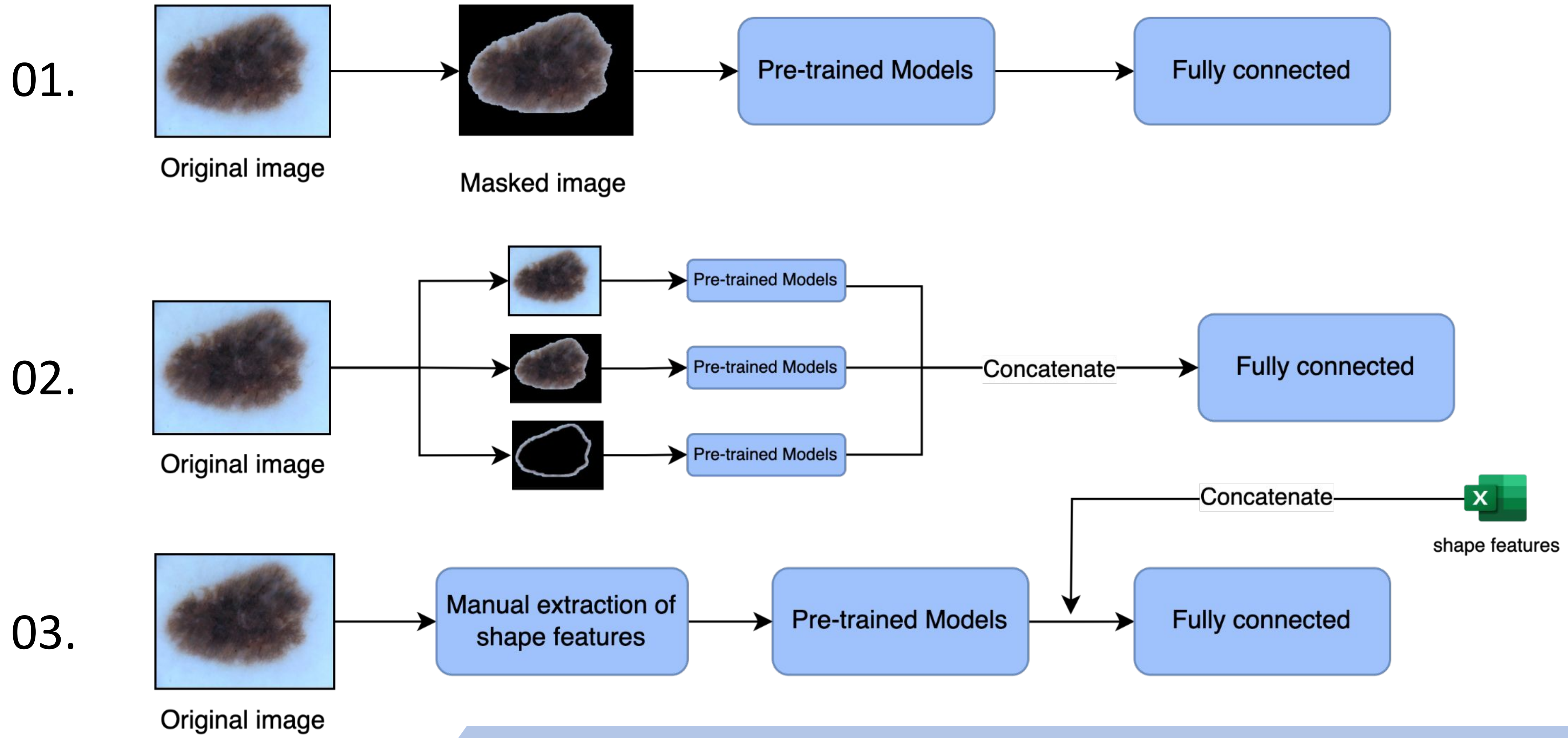


Predicted image

To evaluate the segmentation we used the Dice Score Coefficient (DSC). The final model achieved **95% DSC** on the validation set.

Segmentation Step

Segmentation step:



03

CHALLENGE 1 (BINARY)

Challenge 1: Data Augmentation

Rotation

[0°,90°]

Width Shift

Up to 20% of total width

Height Shift

Up to 20% of total height

Shear Range

[-0.2 rad, +0.2 rad]

Zoom Range

Up to 20%

Horizontal Flip

Brightness Range

By a factor of [0.8,1.2]

Fill Mode

Fill_mode= nearest

Challenge 1: Implementation

Keras

Model	Input Shape	Parameters
ResNet50V2	(224,224,3)	25.6M
EfficientNetV2B2	(260,260,3)	10.2M
EfficientNetB2	(260,260,3)	9.2M
EfficientNetB3	(300,300,3)	12.3M
EfficientNetB4	(380,380,3)	19.5M
EfficientNetB5	(456,456,3)	30.6M
EfficientNetB6	(528,528,3)	43.3M
EfficientNetB7	(600,600,3)	66.7M
DenseNet201	(224,224,3)	14.3M
DenseNet169	(224,224,3)	20.2M

Challenge 1: Training

Framework

- **Keras** and **Tensorflow**

Optimizer

- **Adam optimizer** was used.

Batch Size

- Depending on the network: **16, 32, 64 and 128.**

Transfer learning

- Imagenet weights were used.
- **No layer was frozen.**

Learning Rate

- Initial **lr=1e-4**.
- Reduce on Plateau based on val loss (patient=5).

Loss Function

- **Binary Cross Entropy** (default) was used.

Early Stopping Criteria

- Based on validation loss, with a patient of 10.

04

CHALLENGE 2 (MULTICLASS)

Challenge 2: Data Augmentation

Random Flip

Flip vertically and horizontally by 90°

Random Crop

New cropped area will be a random fraction between 40 - 100 % of original image

Random Affine

[0 - 90°] of rotation
[0 - 20°] of shearing
scaling with [0.8 - 1.2] of the original area

Color Jitter

[0.7 - 1.3] of the original brightness
[0.7 - 1.3] of the original contrast
[0.9 - 1.1] of the original saturation

Challenge 2: Implementation

PyTorch

Model	Input Shape	Parameters
ResNet50	(224,224,3) min	25.56 M
DenseNet161	(256,256,3)	28.68 M
EfficientNetB1	(256,240,3)	7.79 M
Swin Tiny	(224,224,3) min.	28.29 M
Swin Small	(224,224,3) min.	49.61 M
Swin V2 S	(256,256,3)	49.74 M
Swin V2 B	(256,256,3)	87.93 M

Challenge 2: Training

Framework

- **Pytorch**

Optimizer

- **Adam optimizer** was used.

Batch Size

- Depending on the network: **16, and 64.**

Transfer learning

- Imagenet weights were used.
- **No layer was frozen.**

Learning Rate

- Initial **lr=1e-4**.
- Reduce on Plateau based on val loss (patient=7).

Loss Function

- **Categorical Cross Entropy** was used.

Early Stopping Criteria

- Based on validation loss, with a patient of 15.

Training: Class Imbalance

We tried the multiclass cross entropy loss in two different settings:-

Weighting

Original Data Split: Calculates class weights to address class imbalance and subsequently feed to the train model. (mel class = 0.6244), (bcc class = 0.85), (scc class = 4.5053)

Sampling

Balanced Data Split: 1694 samples/class (Sampling with replacement)

05

RESULTS

Evaluation Criteria for Challenges

Binary challenge

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

True Labels

Predicted Labels	Nevus	Other
Nevus	TP	FP
Other	FN	TN

Multiclass challenge

$$kappa(\kappa) = \frac{P_o - P_e}{1 - P_e}$$

$$\textbf{Balanced Accuracy} = \frac{1}{3}\left(\frac{TP}{TP+FN}\right)$$

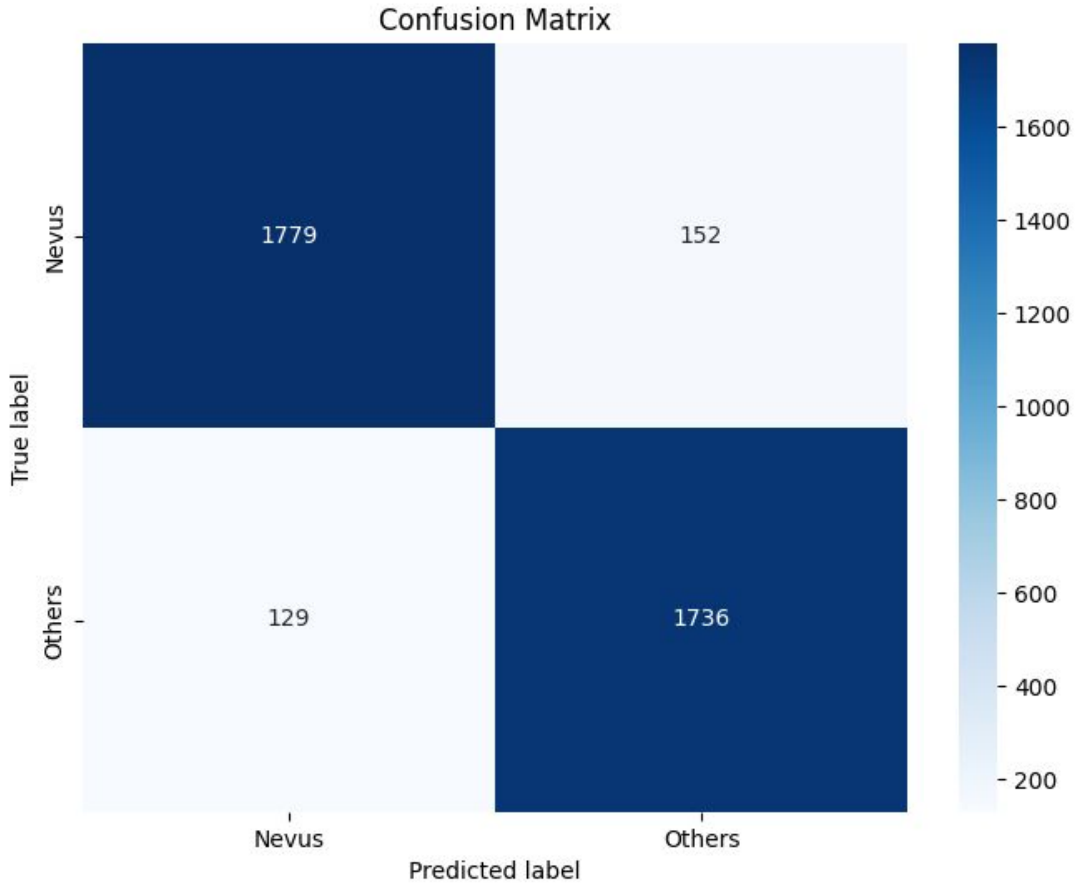
True Labels

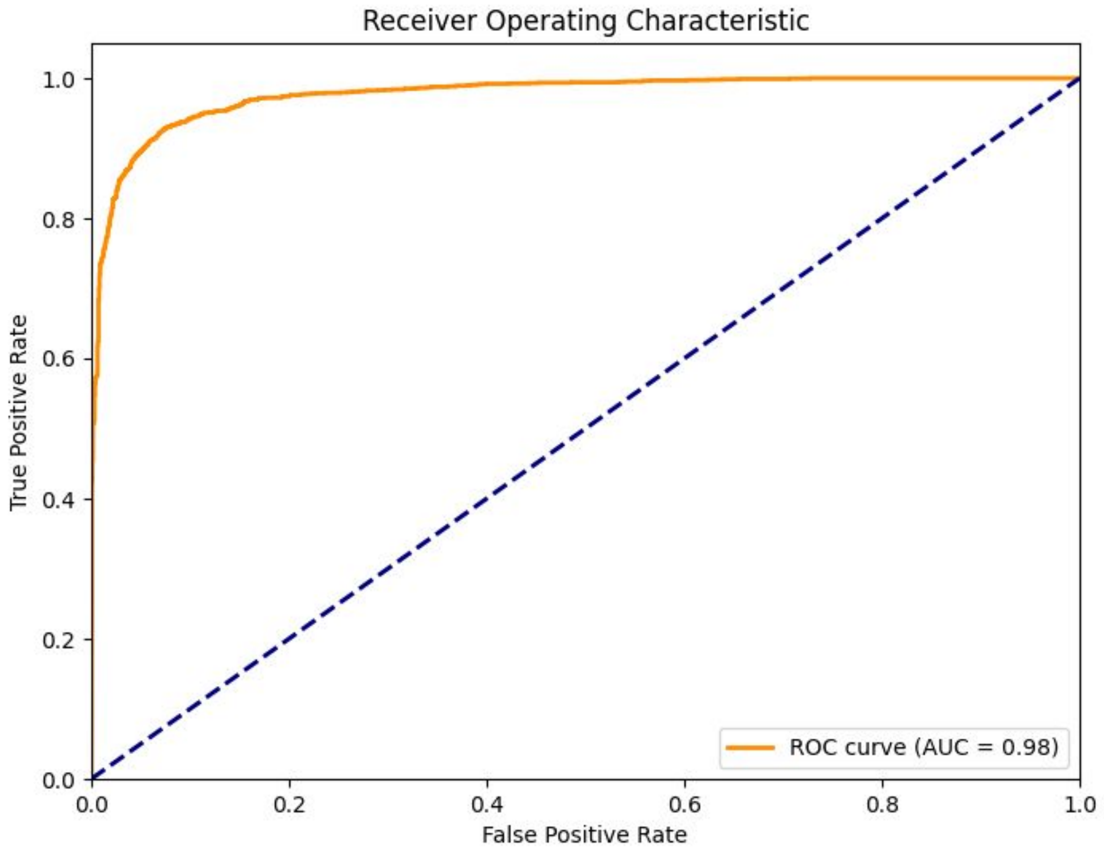
Predicted Labels	Mel	Bcc	Scc
Scc	TN	FP	TN
Bcc	FN	TP	FN
Mel	TN	FP	TN

Results: Challenge 1 (Binary Class)

Model	Accuracy	Kappa
ResNet50V2	0.8895	0.7788
EfficientNetV2B2	0.8871	0.7743
EfficientNetB2	0.8940	0.7879
EfficientNetB3	0.9059	0.8119
EfficientNetB4	0.9186	0.8373
EfficientNetB5	**0.9261**	**0.8521**
EfficientNetB6	0.9245	0.8489
EfficientNetB7	0.8935	0.7866
DenseNet201	0.9028	0.8055
DenseNet169	0.8959	0.7918

Results: Single Best Model (Binary)



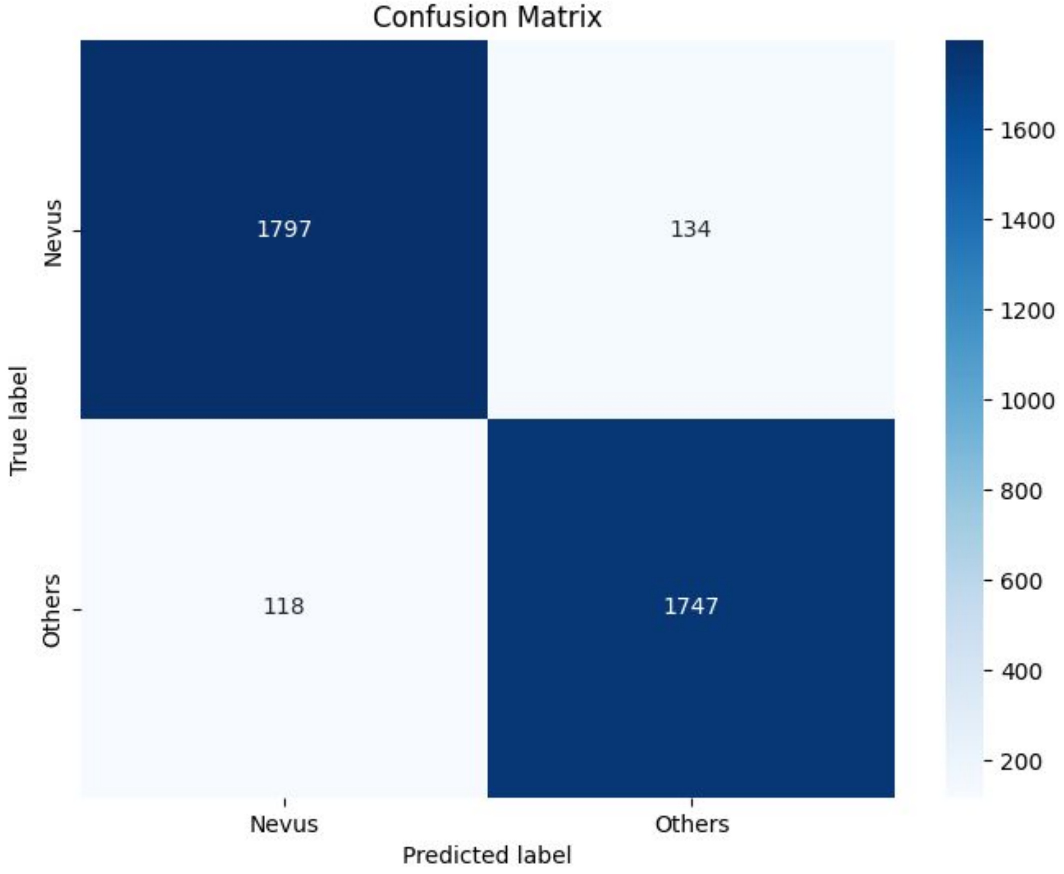


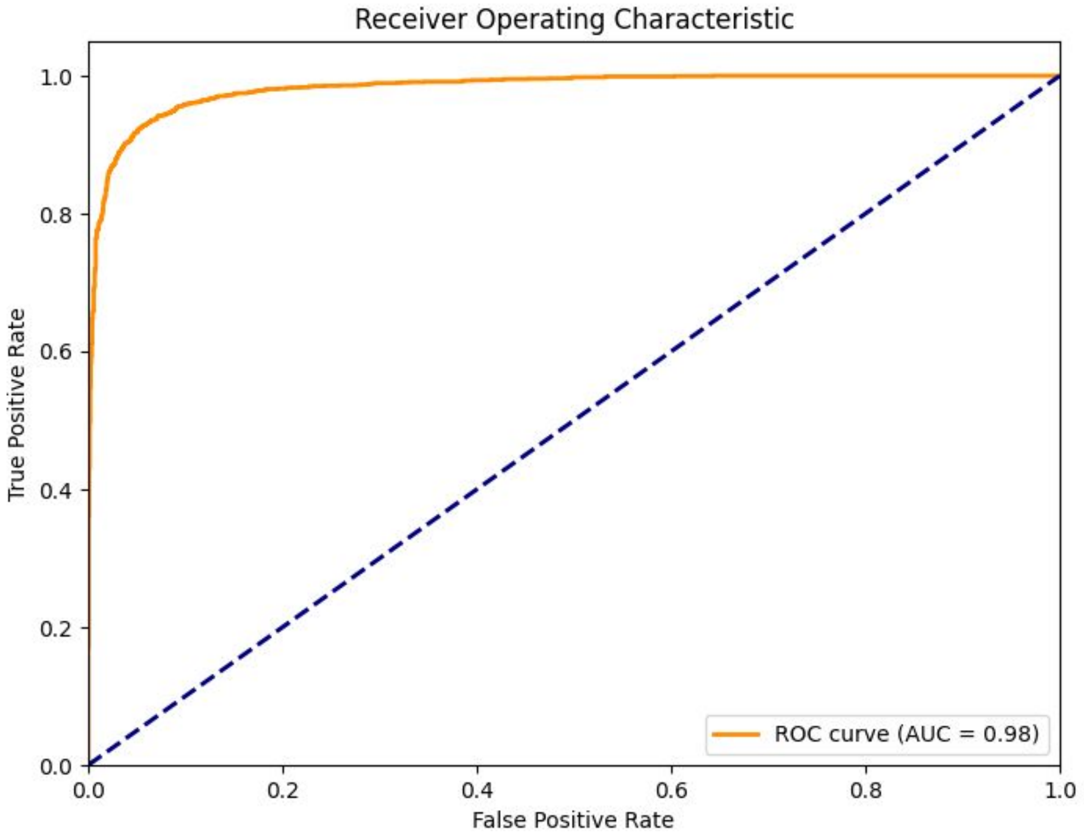
Results: Ensembles Challenge 1 (Binary Class)

	Mean Probabilities		Max Probabilities		Majority Voting	
Models Name	**Accuracy**	**Kappa**	**Accuracy**	**Kappa**	**Accuracy**	**Kappa**
All networks	0.9270	0.8539	0.8464	0.6942	0.8669	0.7327
All EfficientNet	0.9296	0.8592	0.8680	0.7370	0.8790	0.7571
Top 5 results (B3,B4,B5, B6, DenseNet201)	0.9320	0.8640	0.8883	0.7772	0.9051	0.8090
Top 3 results (B4,B5, B6)	**0.9336**	**0.8672**	0.9120	0.8243	0.9212	0.8421
Top 2 results (B5, B6)	0.9317	0.8635	**0.9233**	**0.8468**	**0.9270**	**0.8538**

For the prediction of the test set, we used the ensemble utilising **the top 3 best single accuracies**, which are the EfficientNet B4, B5 and B6.

Results: Best Ensemble Model (Binary)



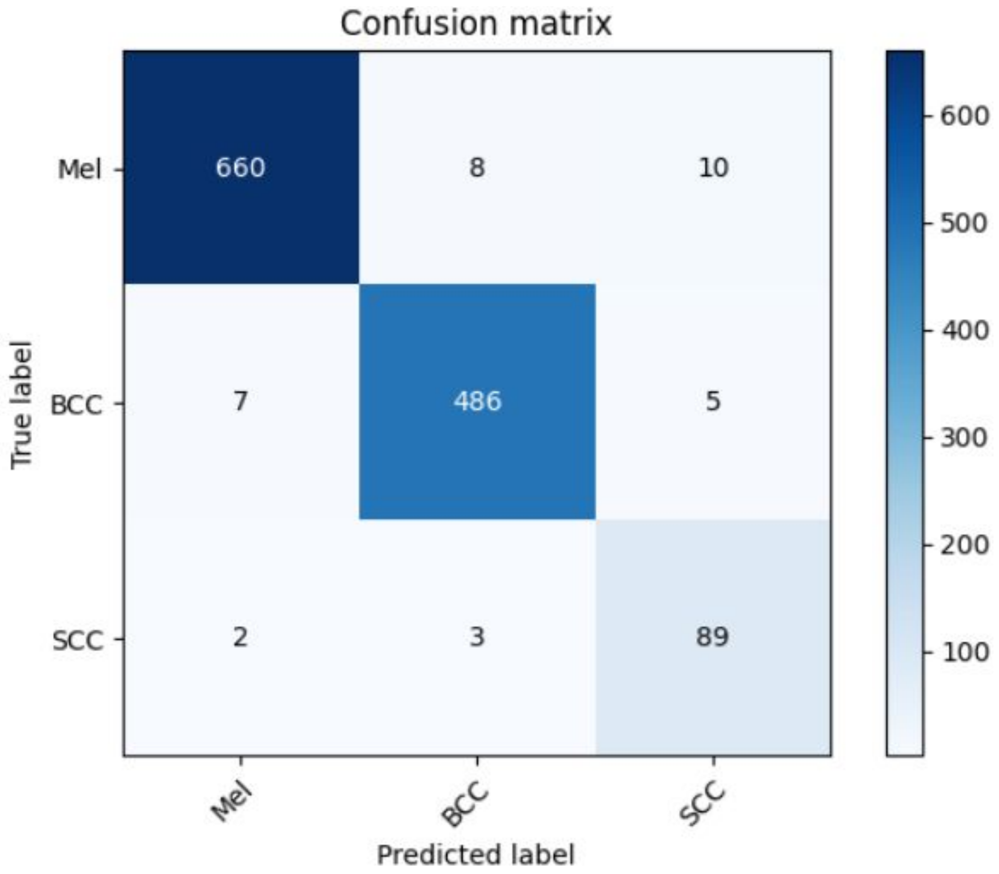


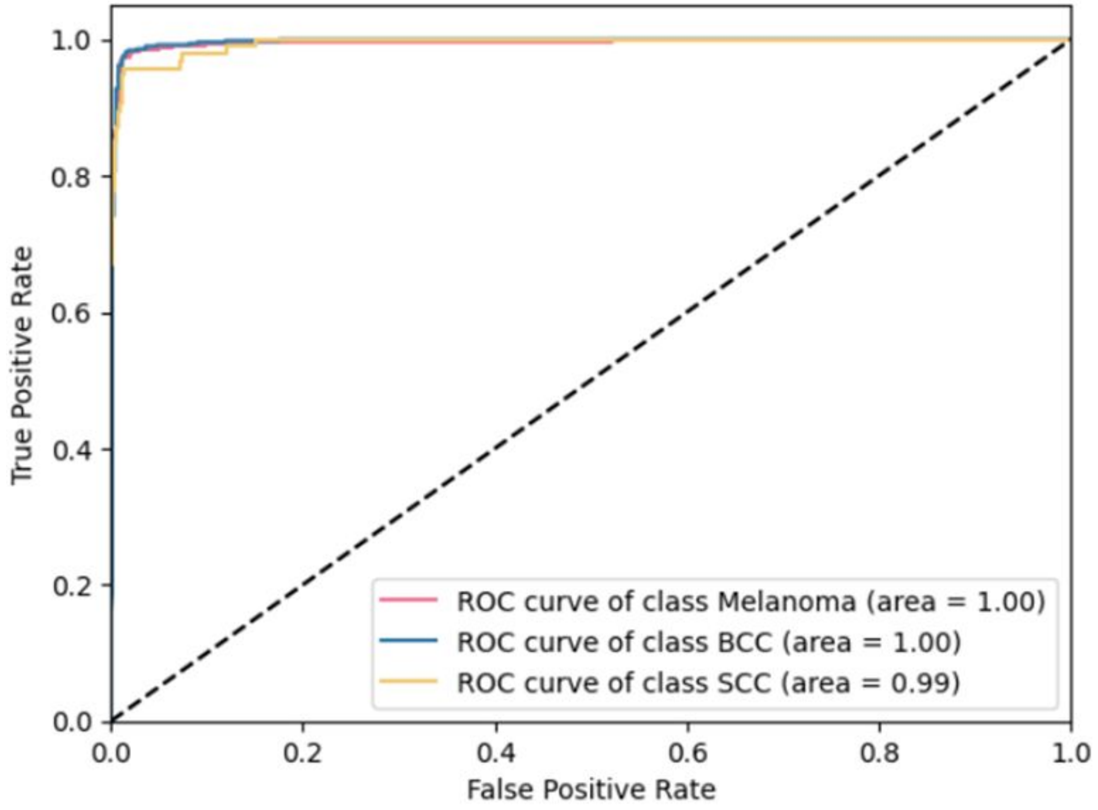
Results: Challenge 2 (Multiclass)

Model	Loss	Accuracy	Kappa
ResNet50	0.1470	0.9598	0.9278
DenseNet161	0.1691	0.9504	0.9108
EfficientNetB1	0.1555	0.9606	0.9293
Swin Tiny	0.1683	0.9630	0.9338
Swin Small	0.1477	0.9598	0.9280
Swin V2 S	**0.1282**	**0.9724**	**0.9507**
Swin V2 B	0.1396	0.9724	0.9504

Results: Best Ensemble Model (Multiclass)

MAIA





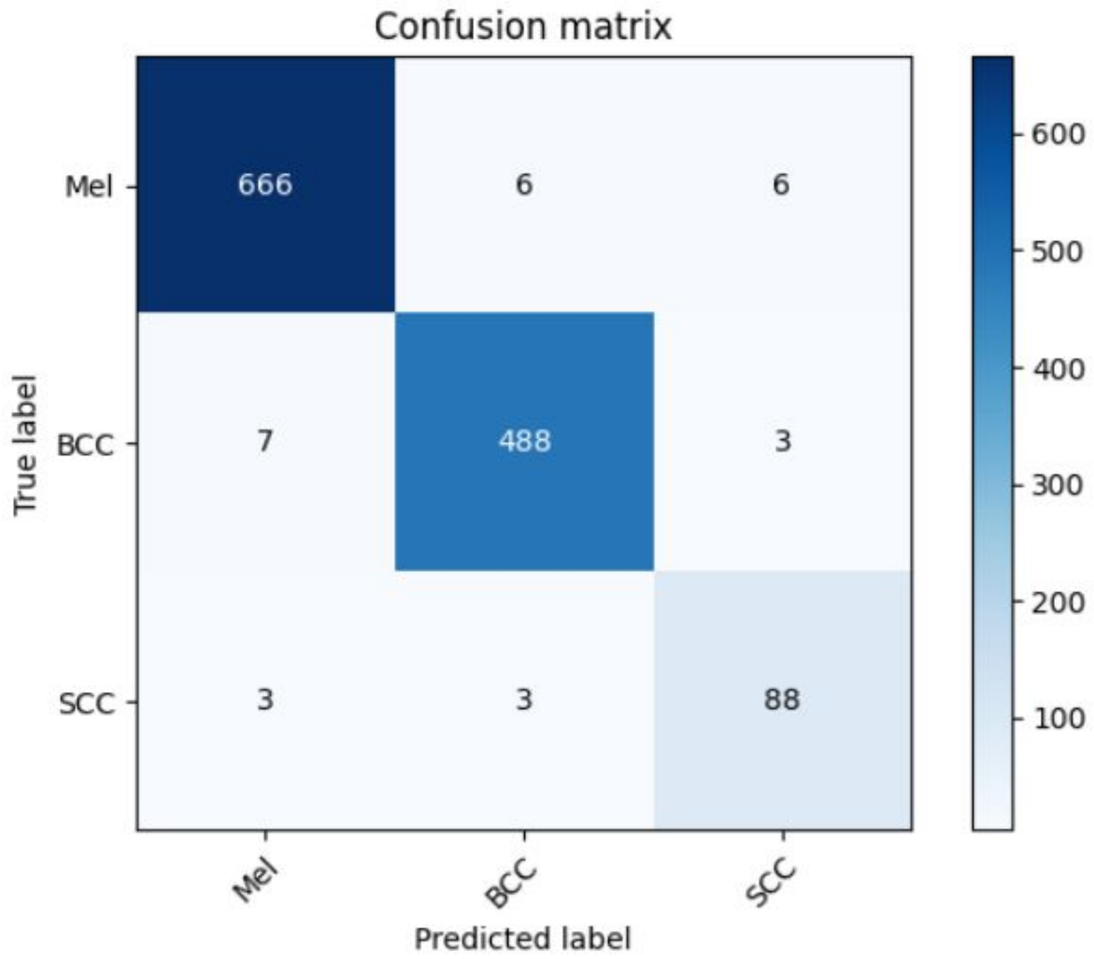
Results: Challenge 2 (Multiclass)

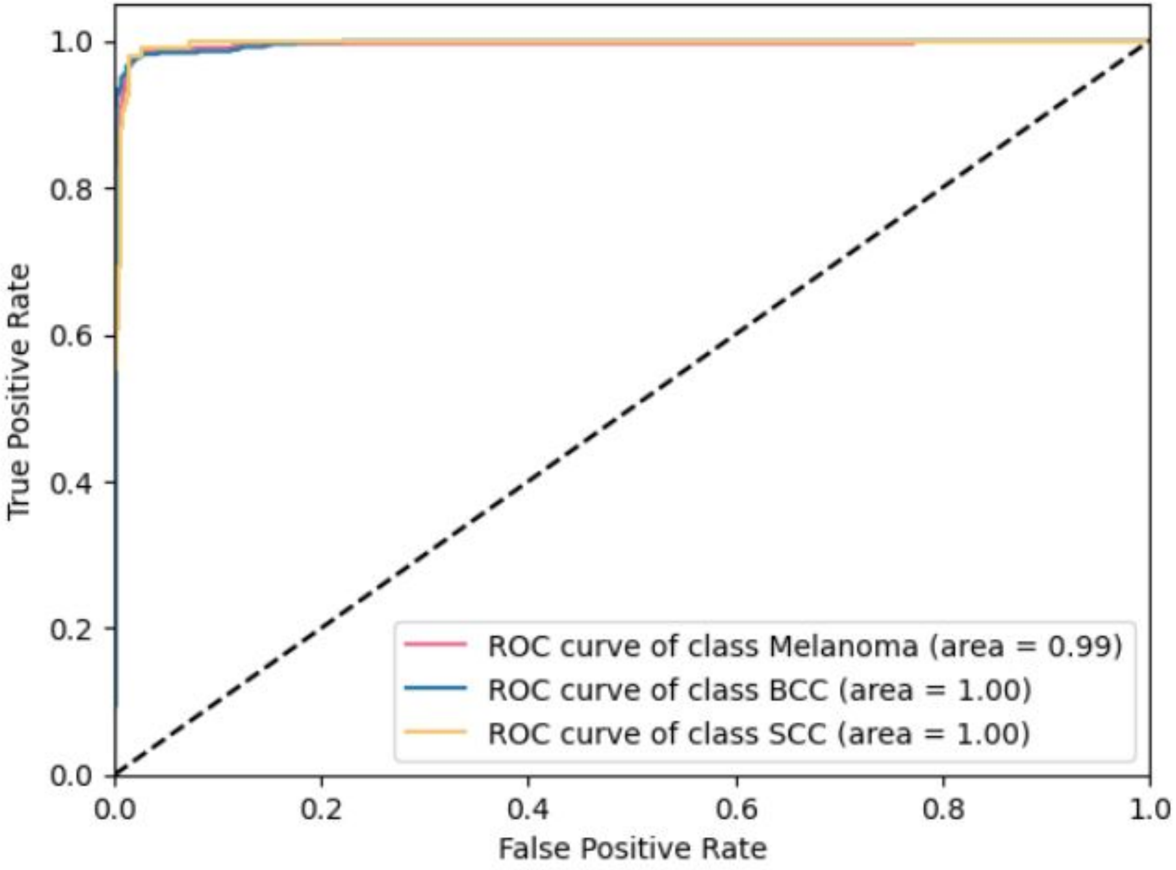
MAIA

Ensemble Models	Accuracy	Kappa
Swin S + Swin V2 S + Swin V2 B	0.9732	0.9520
Swin T + Swin S + Swin V2 S + Swin V2 B	0.9740	0.9534
ResNet50 + Swin S + Swin V2 S + Swin V2 B	0.9748	0.9547
EfficientB1 + Swin S + Swin V2 S + Swin V2 B	**0.9780**	**0.9604**

For the prediction of the test set, we used the ensemble utilising the EfficientNet B1, Swin S, Swin V2 S and Swin V2 B.

Results: Best Ensemble Model (Multiclass)





06

CONCLUSIONS AND FUTURE SCOPE

Conclusion and Future Scope

- Deep Learning outperformed classical approaches in both binary and multiclass challenge.
- Fine-Tuning the hyperparameters of training models is important and challenging.
- Leveraging transfer learning proves highly beneficial for enhancing the performance of Deep Learning models, even when confronted with different datasets.
- Transformers perform very well in Computer Vision and give comparable results to Convolutional Neural Networks.

References

[1] Ha, Qishen & Liu, Bo & Liu, Fuxu. (2020). Identifying Melanoma Images using EfficientNet Ensemble: Winning Solution to the SIIM-ISIC Melanoma Classification Challenge.

[2] Team, K. (n.d.). *Keras documentation: Image classification via fine-tuning with EfficientNet*. https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/

[3] https://pytorch.org/vision/main/models/swin_transformer.html

[4]https://github.com/pytorch/vision/blob/d2bfd639e46e1c5dc3c177f889dc7750c8d137c7/references/classification/train.py#L92-L93

[5] Perez, C. Vasconcelos, S. Avila, and E. Valle, “Data augmentation for skin lesion analysis”, in Or 2.0 context-aware operating theaters, computer assisted robotic endoscopy, clinical image-based procedures, and skin image analysis (Springer, 2018), pp. 303–311.

Thank you