CAD Project II: Deep Learning

Kaouther Mouheb
Rachika Elhassna Hamadache

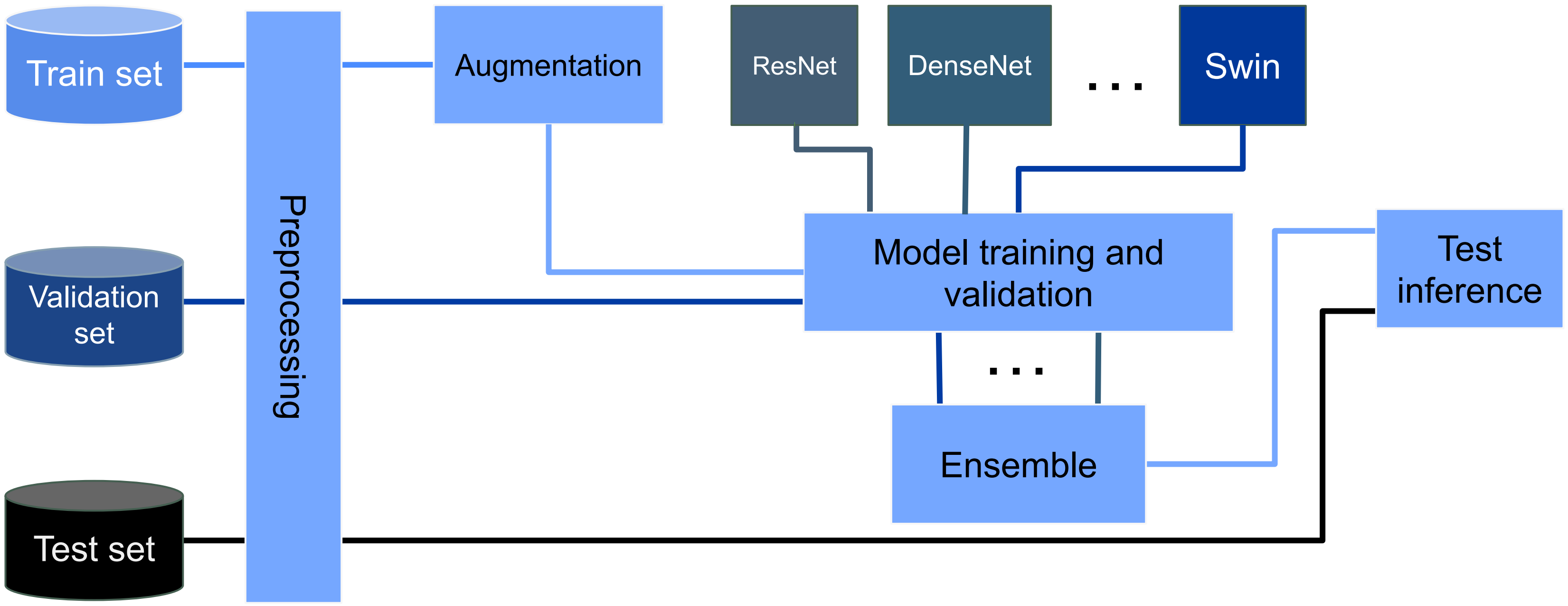
MAIA Fall 2022-2023

Table of Contents

01

Proposal Analysis

Proposed Project Pipeline



02

Implementation

2.1 Data Preparation

1. Data Pre-processing

Resize

- Images are resized to the same size.
- *260x260* pixels.

Center Crop

- Images are center-cropped to the same size.
- *256x256* pixels.

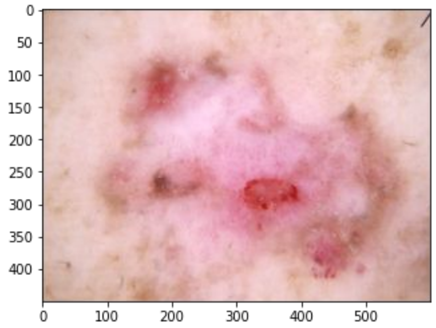
Convert to Tensor

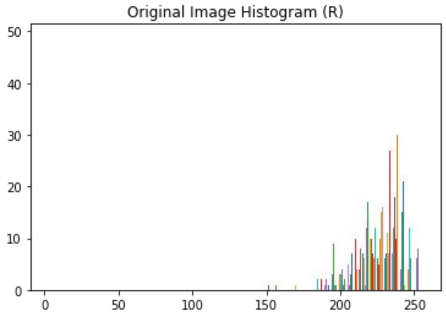
- Images are converted from PILImage to PyTorch's *Tensors*.

Normalize

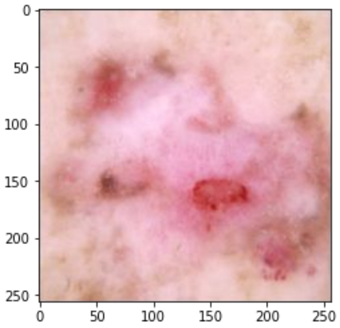
- *Mean* and *standard deviation* are computed from the *train set*.
- *Z-score standardization* is applied.

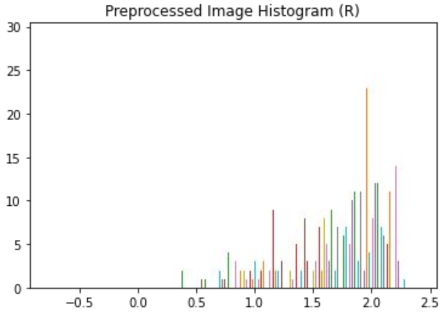
Preprocessing Example





Preprocess





2. Data Augmentation

Following the results presented in [1] the following pipeline was used for data augmentation:

Random Crop → **Random Affine** → **Random Flip** → **Color Jitter**

Random Crop

- [0.4 – 1.0] of the original *area*.
- [3/4 – 4/3] of the original *aspect ratio*.
- P = 0.7.
- *256x256* pixels.

Random Affine

- [0° - 90°] of *rotation*.
- [0° - 20°] of *shearing*.
- *Scaling* with [0.8 - 1.2] of the original *area*.
- P = 0.8.

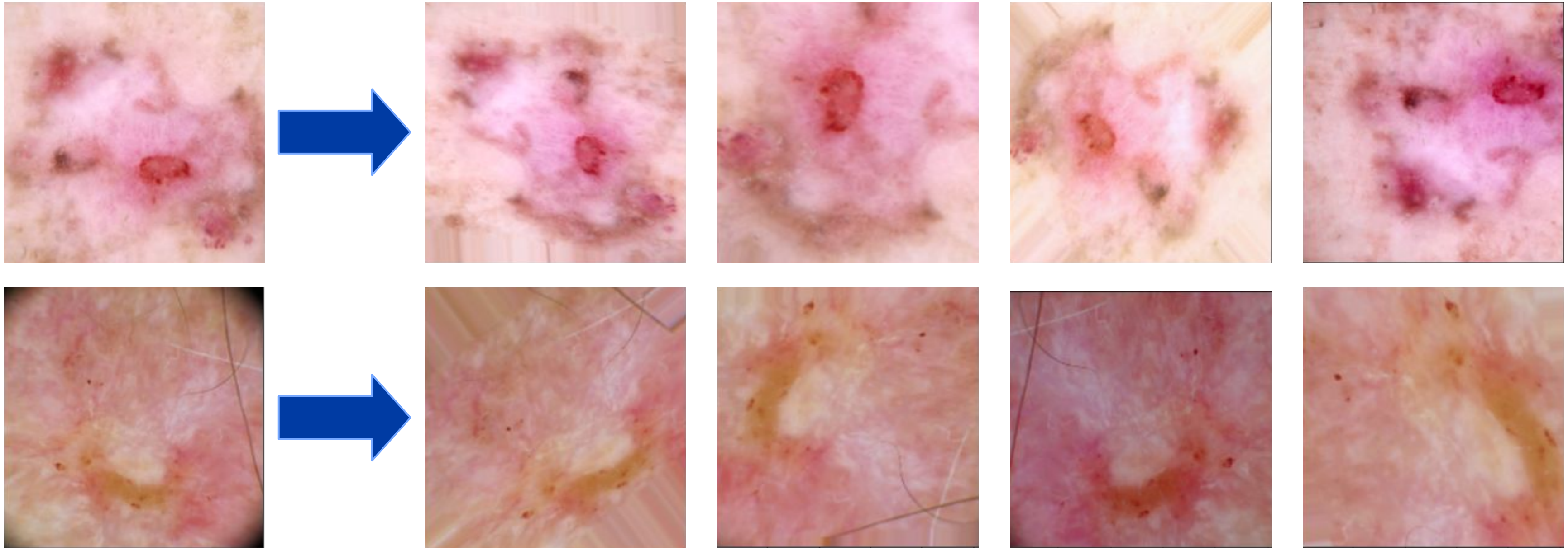
Random Flip

- *Flip vertically* and/or *horizontally*.
- P = 0.9.

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*.
- [-0.05 - 0.05] of the original *hue*.
- P = 0.6.

Augmentation Examples



Original

Augmentation examples

2.2 Training

Implementation details

Training Implementation Details

Framework

- *PyTorch* was used as a framework, with focus on *TorchVision*.

Optimizer

- *Adam* optimizer was used with *weight decay* (10^{-8} ~10^{-6}).

Transfer Learning

- *ImageNet* weights used as initialization, *without freezing* any layers.

Metrics

- Train and validation *losses*, *accuracies* and/or *kappa* scores were monitored.
- Weights providing the *best validation metric* were saved.

Learning Rate

- Small learning rate (10^{-4}) with *ReduceOnPlateau* scheduling based on validation loss.

Early Stopping

- Training was set to stop after a certain number of epochs if the *validation loss does not improve*.

2.3 Challenge I

Binary Classification: Nevus Vs Others

Training Models

Model	Variant
ResNet	• ResNet34 • ResNet50
ResNeXt	• ResNeXt50
SwinTransformer	• SwinTransformerT (Tiny) • SwinTransformerB (Base)
EfficientNet	• EfficientNetB2 • EfficientNetB3 • EfficientNetB4 • EfficientNetB5 • EfficientNetB6

Training Details

Network Architectures	• ResNet34 • ResNet50 • ResNeXt50 • SwinTransformerT • EfficientNet B2, B3	• SwinTransformerB • EfficientNet B4 - B6
Training batch size	32	4~16
Validation batch size	64	
Learning rate scheduling patience	7	
Early stopping patience	15	
Criterion	Binary Cross-Entropy loss	

2.4 Challenge II

Multi-Class Classification: Melanoma Vs BCC Vs SCC

Training Strategies

1. Multi-Class Focal Loss

- We used the *multi-class focal loss* as an *objective function* to optimize the models as introduced in [2] which is given below:

$$\mathfrak{L}f(p_t) = -\alpha(1 - p_t)^{\gamma}\log(p_t)$$

- Where:
    - p_t is the model's estimated *probabilities* vector.
    - γ is the *focusing parameter* (set to 2).
    - α is the *class weight* given by:

$$\alpha_c = \frac{N}{C \times n_c} \Rightarrow \alpha_{mel} = 0.6244,\ \alpha_{BCC} = 0.8500,\ \alpha_{SCC} = 4.5053$$

Training Strategies

2. Multi-Class Cross Entropy Loss

- We used the *multi-class cross entropy loss* in three different settings:

Standard	$\mathcal{L}_{CE} = -log(p_t)$		Original Data Split
Weighting	$\mathcal{L}_{WCE} = -\alpha_t log(p_t)$	α_{mel}= 0.6244 α_{BCC}= 0.8500 α_{SCC}= 4.5053	Original Data Split
Sampling	$\mathcal{L}_{CE} = -log(p_t)$		Balanced Data Split 1694 samples/class (Sampling with replacement)

Training Details

Network Architectures	• ResNet50 • DenseNet161 • EfficientNetB2 • SwinTransformer (Tiny)
Training batch size	16
Validation batch size	64
Learning rate scheduling patience	7~10
Early stopping patience	10~15

03

Results

3.1 Challenge I

Binary Classification: Nevus Vs Others

Single Models

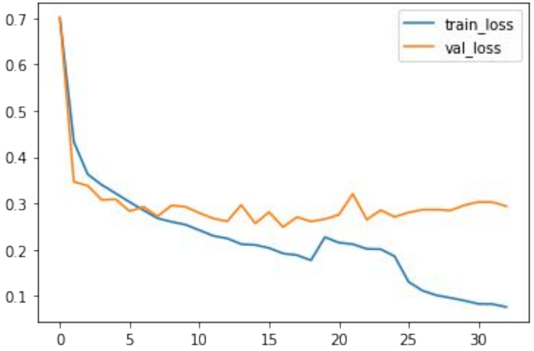
Model	Variant	Loss	Accuracy	Kappa
ResNet	ResNet34	0.3161	0.9023	0.8044
	ResNet50	0.3476	0.9093	0.8186
ResNeXt	ResNeXt50	0.3037	0.9131	0.8260
SwinTransformer	Swin Transformer T	0.3034	0.9056	0.8113
	Swin Transformer B	**0.3026**	**0.9188**	**0.8377**

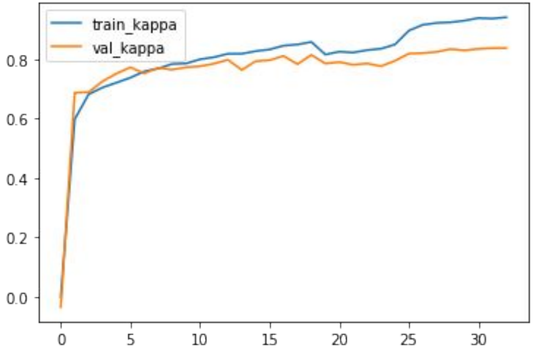
Single Models

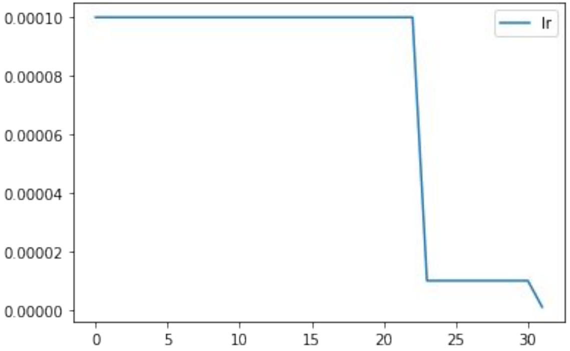
Model	Variant	Loss	Accuracy	Kappa
EfficientNet	EfficientNet B2	0.3039	0.9120	0.8239
	EfficientNet B3	0.2841	0.9106	0.8213
	EfficientNet B3 *(scratch)*	0.3137	0.8722	0.7442
	EfficientNet B4	0.2998	0.9141	0.8281
	EfficientNet B5	0.3619	0.9051	0.8102
	EfficientNet B6	0.3032	0.8914	0.7828

Best Single Model: Swin Base

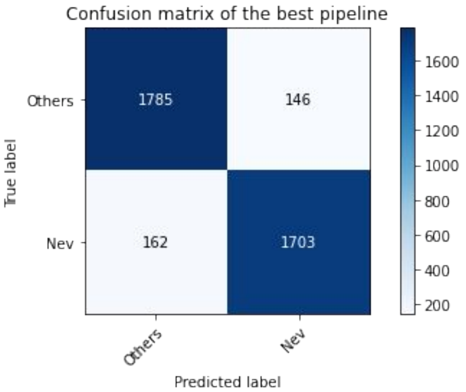
Training

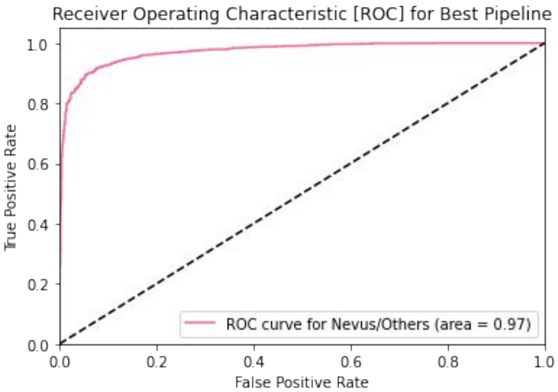






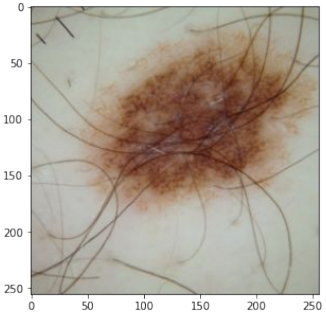
Validation



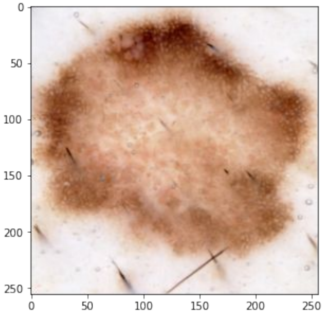


Best Single Model: Swin Base

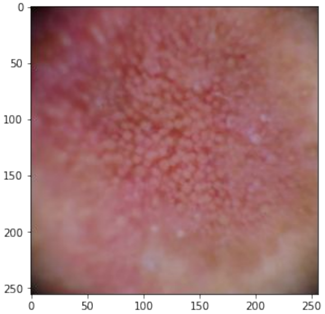
Example Predictions



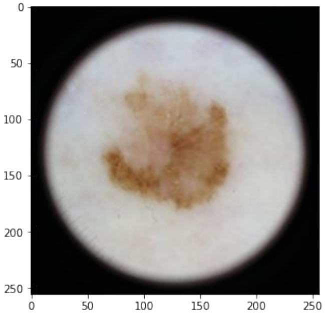
Nev (Nev)



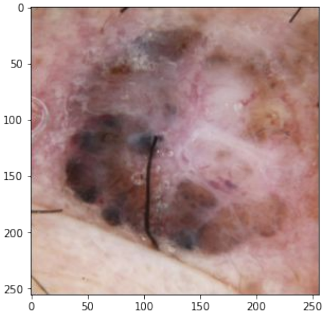
Others (Nev)



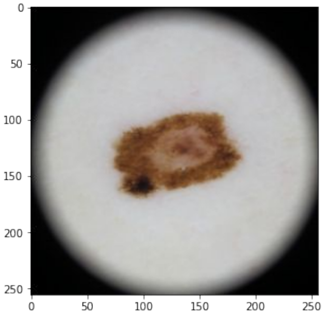
Others (Others)



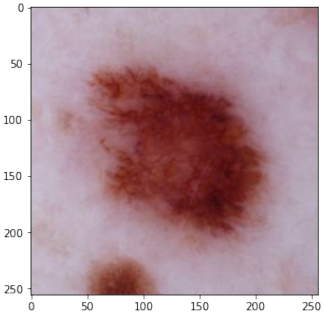
Nev (Others)



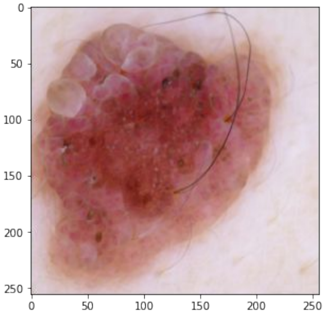
Others (Others)



Nev (Nev)



Others (Nev)



Nev (Nev)

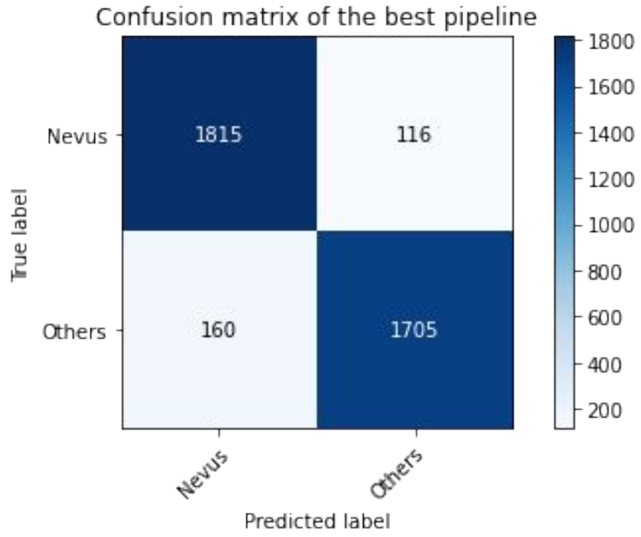
Improvements: Ensembles (Soft)

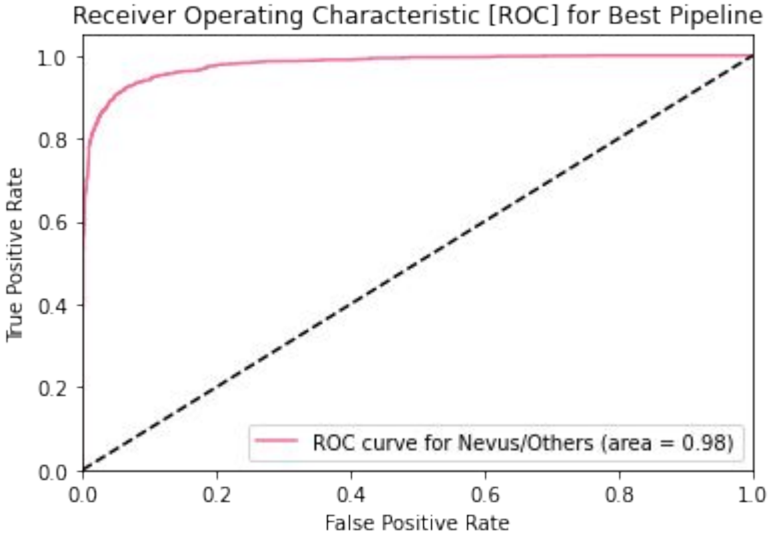
Ensemble	Accuracy	Kappa
SwinB + EffiNetB4	0.9189	0.8377
SwinB + ResNeXt50	0.9262	0.8524
SwinB + EffiNetB4 + ResNeXt50	0.9231	0.8461
SwinB + ResNeXt50 + EffiNetB3	**0.9273**	**0.8545**
SwinB + ResNeXt50 + EffiNetB2	0.9257	0.8513
SwinB + ResNeXt50 + EffiNetB3 + EffiNetB2	0.9268	0.8534
SwinB + SwinT + ResNeXt50 + EffiNetB4 + EffiNetB3 + EffiNetB2	0.9265	0.8529
SwinB + ResNeXt50 + EffiNetB3 + ResNet50	0.9252	0.8503

Final Pipeline

Ensemble of Swin_B + ResNeXt50 + EfficientNet_B3 with soft voting (Average probability)

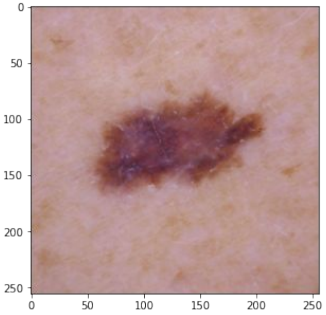
Validation



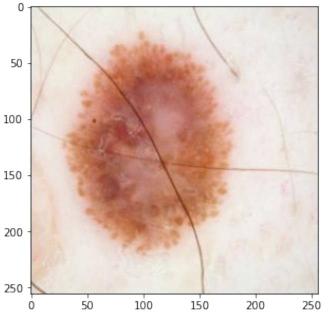


Final Pipeline

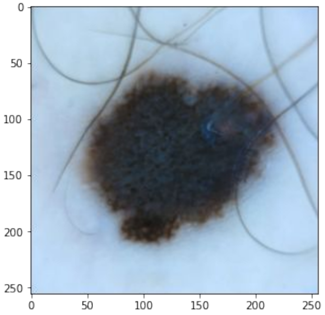
Example Predictions



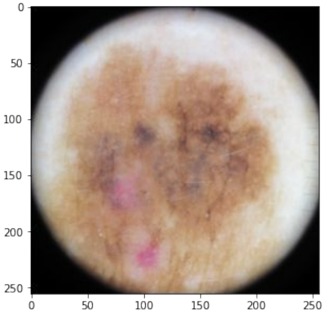
Others (Others)



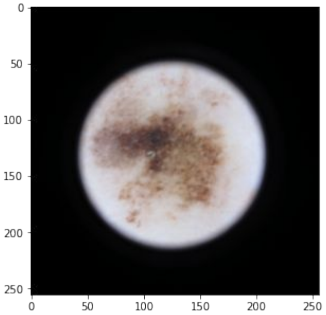
Others (Nev)



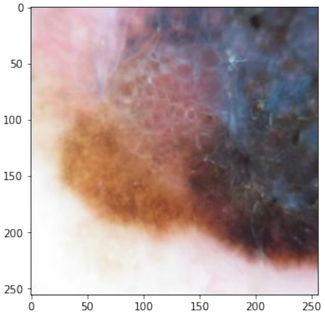
Nev (Nev)



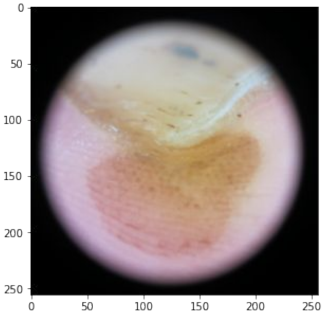
Others (Others)



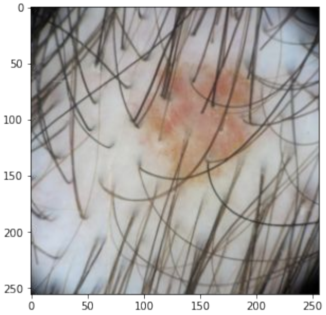
Nev (Others)



Others (Others)



Nev (Nev)



Nev (Nev)

3.2 Challenge II

Multi-Class Classification: Melanoma Vs BCC Vs SCC

Single Models

Model	Train Strategy	Loss	Accuracy	Kappa
ResNet50	Focal Loss	0.1246	0.9244	0.8657
	CE	0.1706	0.9567	0.9219
	Weighted CE	0.1966	0.9472	0.9059
	CE + Sampling	0.2067	0.9543	0.9177
DenseNet161	Focal Loss	0.0982	0.9394	0.8926
	CE	0.1760	0.9543	0.9175
	Weighted CE	0.2027	0.9480	0.9065
	CE + Sampling	0.1813	0.9520	0.9135

Single Models

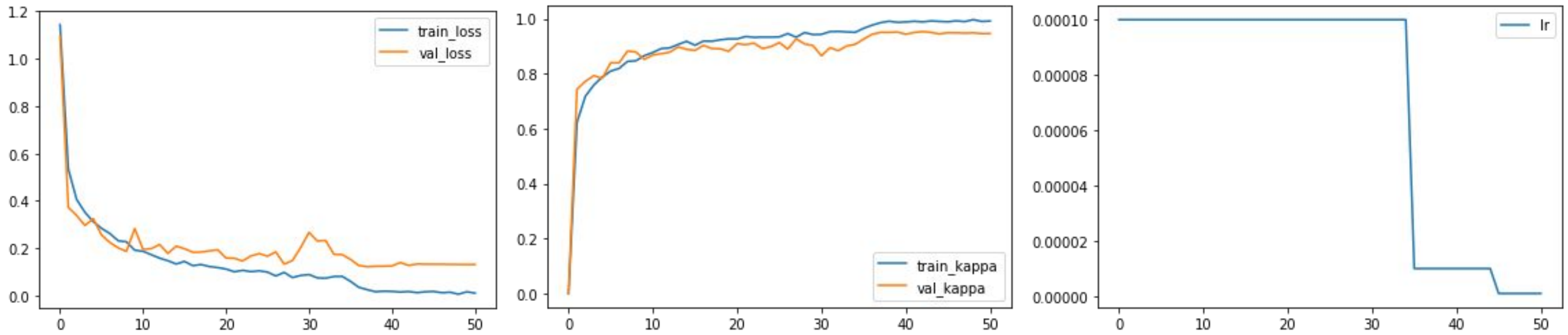
Model	Train Strategy	Loss	Accuracy	Kappa
SwinTransformer (Tiny)	Focal Loss	0.0907	0.9323	0.8799
	CE	**0.1458**	**0.9669**	**0.9406**
	Weighted CE	0.1907	0.9528	0.9153
	CE + Sampling	0.1579	0.9622	0.9320
EfficientNet (B2)	Focal Loss	0.0969	0.9504	0.9112
	CE	0.1706	0.9496	0.9097
	Weighted CE	0.2033	0.9512	0.9131
	CE + Sampling	0.1706	0.9496	0.9097

Improvements: SwinTransformer

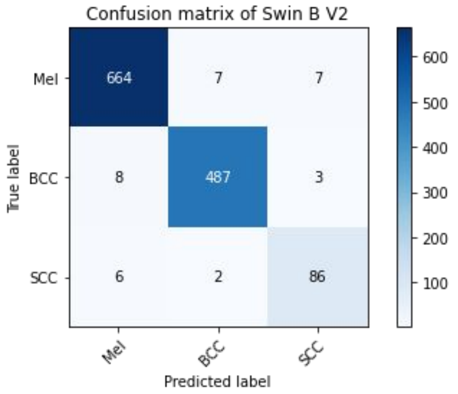
Modification	Loss	Accuracy	Kappa
Baseline (Previous best model)	0.1458	0.9669	0.9406
Swin_S	0.1704	0.9685	0.9431
Swin_S_V2	0.1313	0.9709	0.9476
Swin_B_V2	**0.1281**	**0.9740**	**0.9533**
Swin_B_V2 + weight decay 10^{-6}	0.1700	0.9598	0.9279
Swin_B_V2 + weight decay 10^{-9}	0.1556	0.9661	0.9390
Swin_B_V2, 5 LR 15 ES patience	0.1583	0.9606	0.9290
Test Time Augmentation	-	0.9732	0.9518

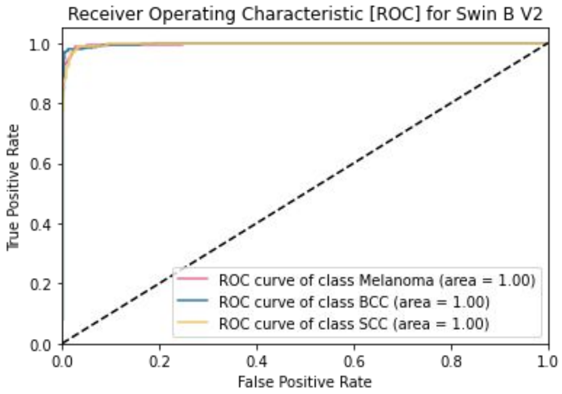
Best Single Model: Swin Base V2

Training



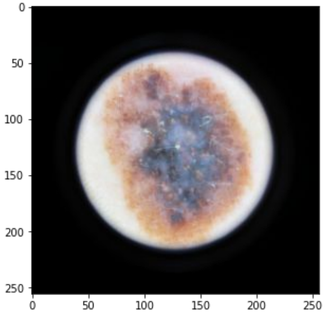
Validation



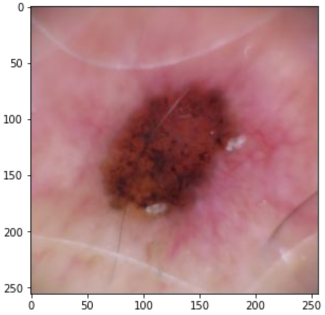


Best Single Model: Swin Base V2

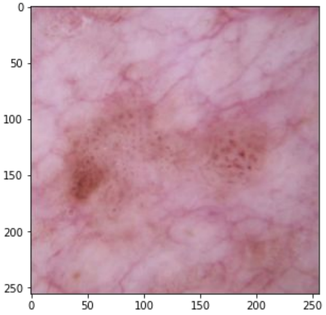
Example Predictions



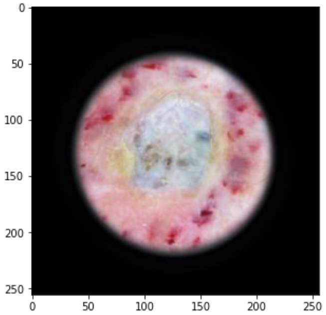
Melanoma (Mel)



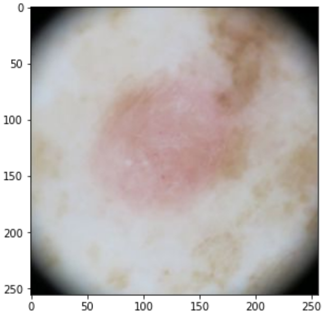
BCC (BCC)



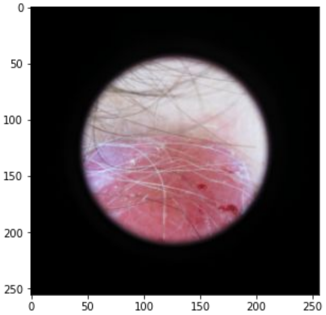
SCC (SCC)



Melanoma (BCC)



SCC (Mel)



SCC (Mel)

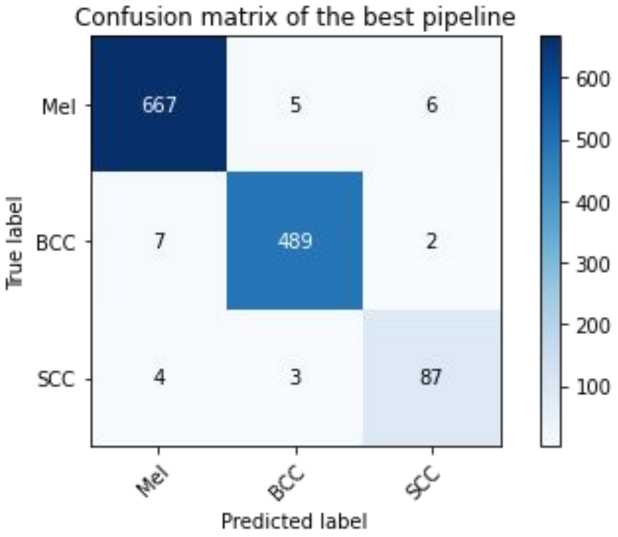
Improvements: Ensembles (Soft)

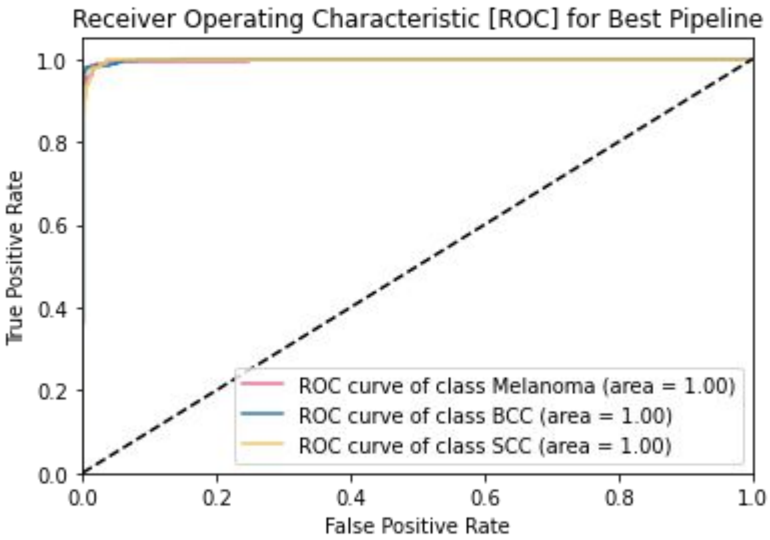
Models	Accuracy	Kappa
Swin_S + Swin_S_V2 + Swin_B_V2 + Swin_T	0.9764	0.9575
Swin_S + Swin_S_V2 + Swin_B_V2	0.9780	0.9603
Swin_S_V2 + Swin_B_V2	0.9756	0.9561
Swin_S_V2 + Swin_B_V2 + EfficientNet_B2	**0.9787**	**0.9618**
Swin_S + Swin_S_V2 + Swin_B_V2 + EfficientNet_B2	0.9772	0.9589
Swin_S_V2 + Swin_B_V2 + ResNet50	0.9764	0.9575
win_S + Swin_S_V2 + Swin_B_V2 + ResNet50	0.9780	0.9603

Final Pipeline

Ensemble of Swin_S_V2 + Swin_B_V2 + EfficientNet_B2 with soft voting (Average probability)

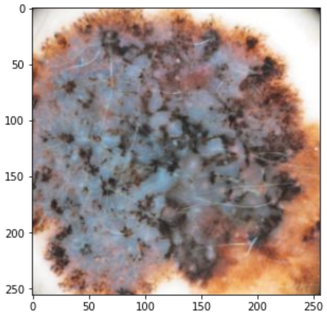
Validation



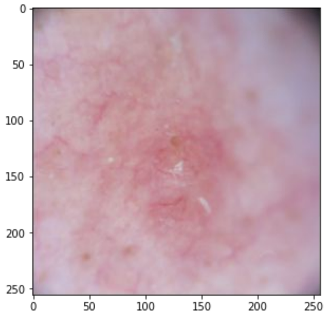


Final Pipeline

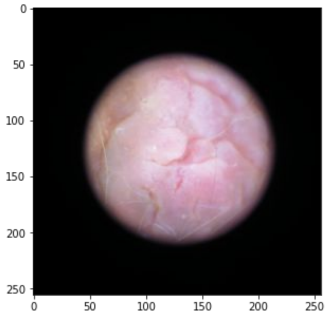
Example Predictions



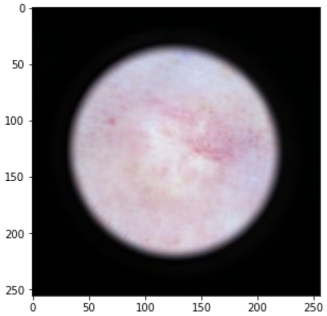
Melanoma (Mel)



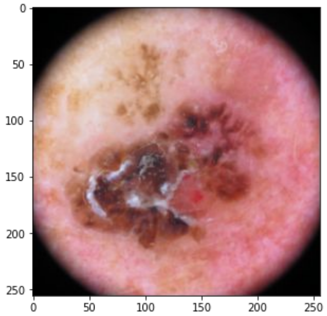
BCC (BCC)



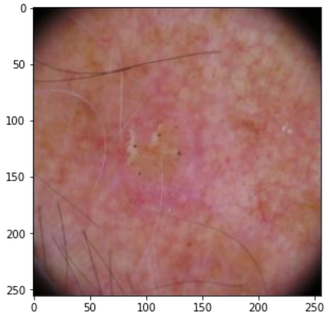
SCC (SCC)



BCC (Mel)



Melanoma (SCC)



SCC (Mel)

04

Conclusion

Conclusion

- Deep learning outperformed classical approaches in both challenges.
- Fine-tuning the *hyperparameters* of training models is *important* and *challenging*.
- *Transfer learning* is very useful in improving the results of Deep Learning models even if the dataset is *different*.
- Deep Learning is *robust* against *class-imbalance* but more improvement could be applied.
- *Transformers* perform very well in Computer Vision and give *comparable* results to *Convolutional Networks*.

Resources

[1] F. 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.

[2] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P.Doll ár, “Focal loss for dense object detection”, in Proceedings of the ieee international conference on computer vision (2017), pp. 2980–2988.

Thank You!