Classifying skin lesions

By Mohamed Elghetany, Megan Martin, Mike Khor

Background & Objective

- Skin cancer is the most common cancer worldwide[1]
- *Early* and *correct* classification impacts treatment
- Lesion appearance is main driver of biopsy/treatment

Objective:
Train a model to classify skin lesion images for different types of cancers and benign lesions

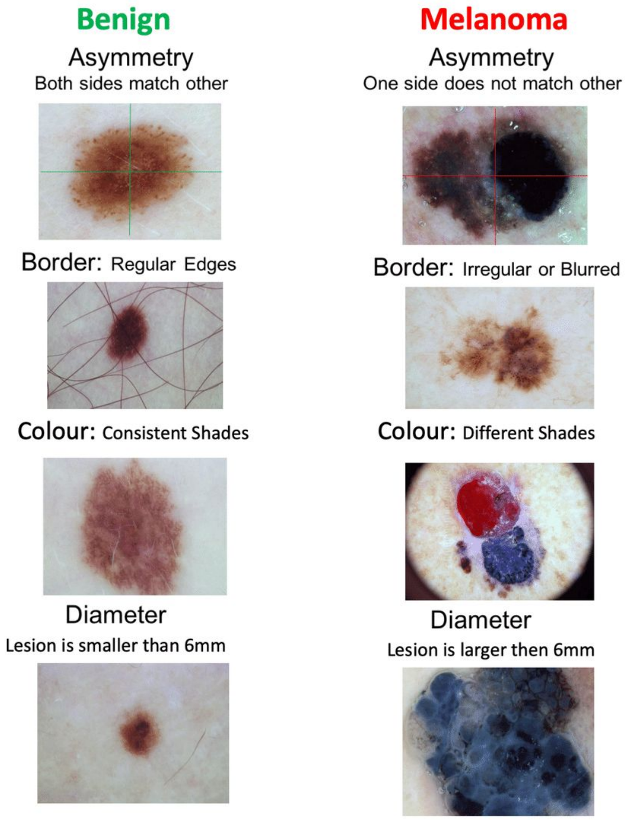


Image obtained from reference 2

About the Data

- Dataset obtained from International Skin Imaging Collaboration (ISIC)[3]
- Utilizing ISIC 2019 dataset which contains dermoscopic images for **8** diagnostic categories
  - Patient metadata for each image
- Dataset contains **25,331** images
  - 21,311 images with complete metadata
  - **Unbalanced** dataset - most are labeled as benign

	image	age_approx	anatom_site_general	lesion_id	sex
0	ISIC_0000000	55.0	anterior torso	NaN	female
1	ISIC_0000001	30.0	anterior torso	NaN	female
2	ISIC_0000002	60.0	upper extremity	NaN	female
3	ISIC_0000003	30.0	upper extremity	NaN	male
4	ISIC_0000004	80.0	posterior torso	NaN	male
...	...	...	...	...	...
25326	ISIC_0073247	85.0	head/neck	BCN_0003925	female
25327	ISIC_0073248	65.0	anterior torso	BCN_0001819	male
25328	ISIC_0073249	70.0	lower extremity	BCN_0001085	male
25329	ISIC_0073251	55.0	palms/soles	BCN_0002083	female
25330	ISIC_0073254	50.0	upper extremity	BCN_0001079	male

25331 rows × 5 columns

Metadata for ISIC 2019 dataset.

Addressing Class Imbalance

MEL	NV	BCC	AK	BKL	DF	VASC	SCC
4522.0	12875.0	3323.0	867.0	2624.0	239.0	253.0	628.0

- >50% images in a single class (benign moles)
- Downsampled or upsampled classes to 2,000 images each, resulting in 16,000 total images for balanced dataset
- Augmentation including random flips, rotations, saturation, contrast, brightness changes were utilized to upsample

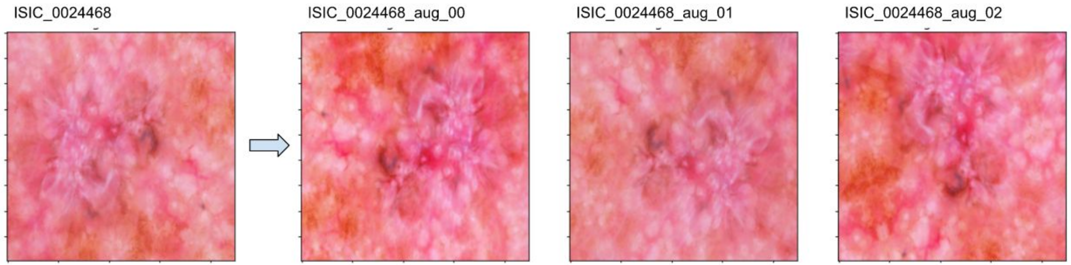
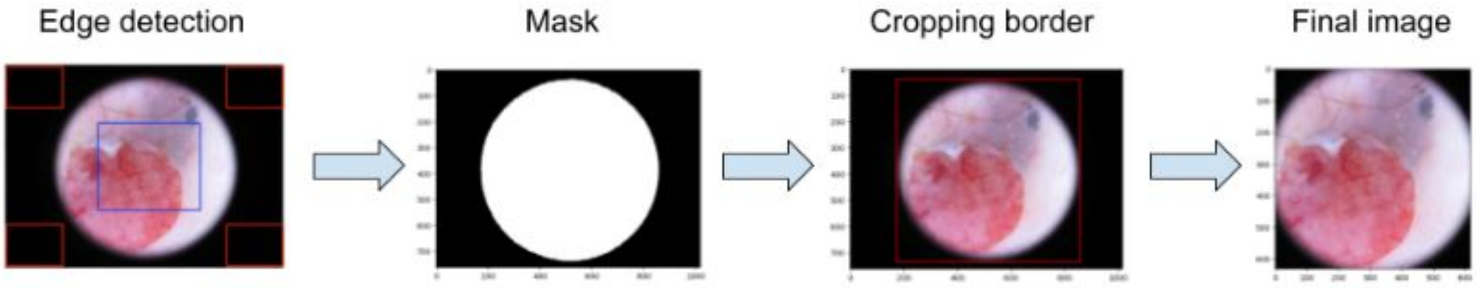
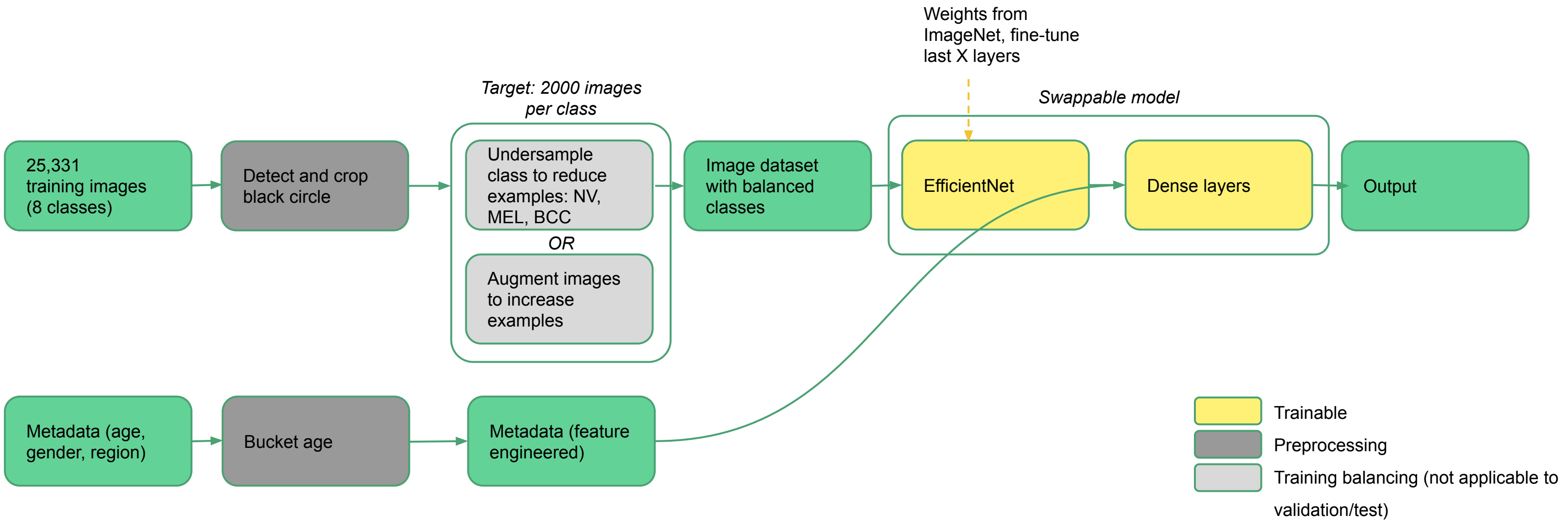


Image pre-processing

- Scaled down varied image sizes to 224 x224
- 25% of images had black regions surrounding lesion
    - *Detection*: does this image need cropping? Use value difference between corner and center.
    - *Mask*: find all pixels that are above a threshold value – “light region”
    - *Determine crop border*: min/max position of “light region”
    - *Crop*: produce final image



Block Diagram



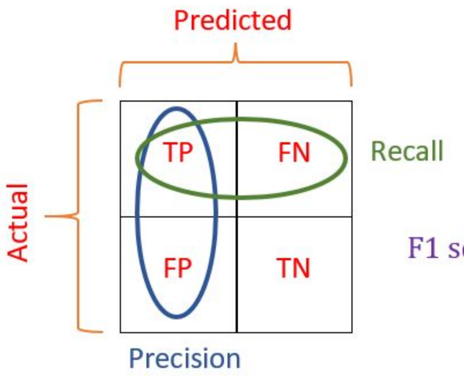
Tech Stack

- Jupyter Lab in Amazon Sagemaker (varying EC2 sizes)
- Python 3 image: Tensorflow 2.6, Python 3.8 GPU optimized
- Scikit-Learn
- Keras
- Numpy, Matplotlib, Pandas

Evaluation and Success Parameters

Success defined as:

- Recall: > 60%
- Precision: >50%
- ROC AUC: >60%
- Accuracy: >55%



$$\text{F1 score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}}$$

Models and Methods

1. Logistic Regression
2. K-Nearest Neighbors (KNN)
3. Random Forest
4. Convolutional Neural Network (CNN)
5. CNN + hyperparameter tuning using Optuna
6. CNN + transfer learning using EfficientNet
7. CNN + Optuna + EfficientNet
8. CNN + XGBoost
9. CNN+ EfficientNet + Metadata (mixed image and structured data)
10. Clustering of Meta data using K-means and K-prototypes

Logistic Regression, KNN, Random Forest

First, we tried standard algorithms- results were not promising

- Logistic regression: noisy validation results
- K-nearest neighbor: many false positives for MEL
- Random forest: n=3 (best accuracy), also false positives for MEL

Figure: Training history of logistic regression

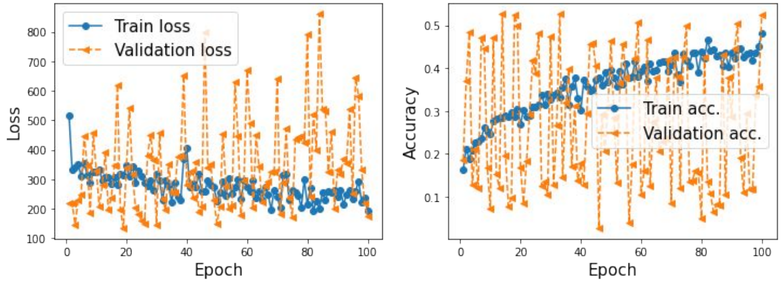


Figure: KNN accuracies across k's

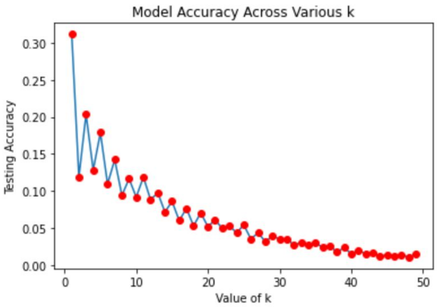
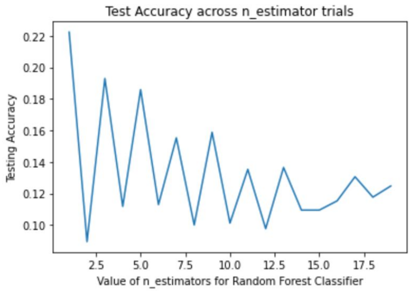


Figure: Random forests across number of trees



CNN

Brief description of approach

1. Convolutional Neural Network (CNN)
2. CNN + hyperparameter tuning using Optuna
3. CNN + transfer learning using EfficientNet
4. CNN + Optuna + EfficientNet
5. CNN + XGBoost
6. CNN+ EfficientNet + Metadata (mixed image and structured data)

OPTUNA

Figure 5. Hyperparameter tuning of CNN structure

1. *A* sets of: (options: 2, 3)
   - Conv2D with:
     - *B* filters (options: 5 to 16)
     - *C* kernel size (options: 3 to 5)
     - *D* setting of padding/no padding (options: True/False)
     - activation: relu
   - MaxPooling
     - *E* pool size (options: 2, 3)
     - *F* stride (options: 2, 3)
   - Dropout: 0.1
2. Flatten
3. *G* layers of Dense (options: 0 to 3)
   - *H* units (options: 5 to 20)
   - Activation: relu
   - Dropout: 0.1
4. One output layer of Dense
   - output units: 8
   - Activation: softmax

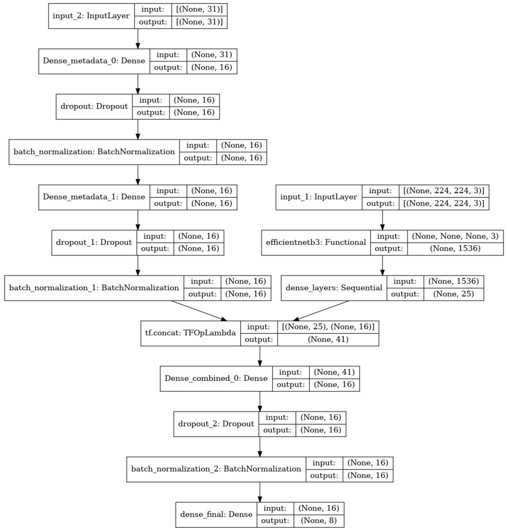
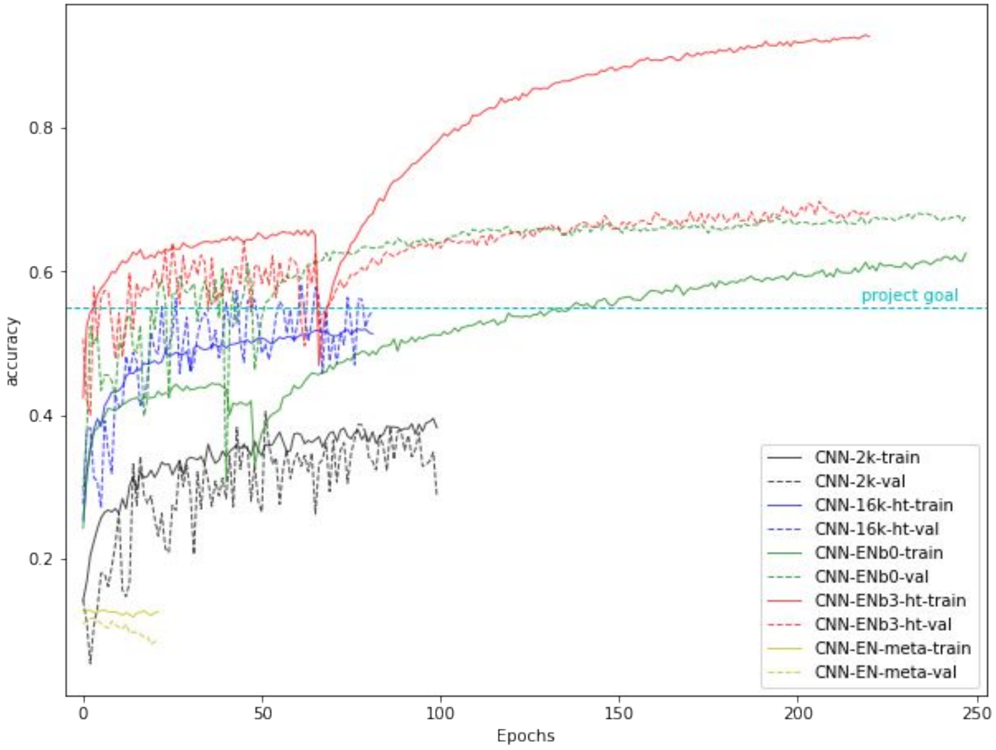


Figure: CNN + metadata input (normal dense layers; left) and image data input (CNN; right), which gets concatenated and put through a dense layer before classification.

Accuracy: top CNN models

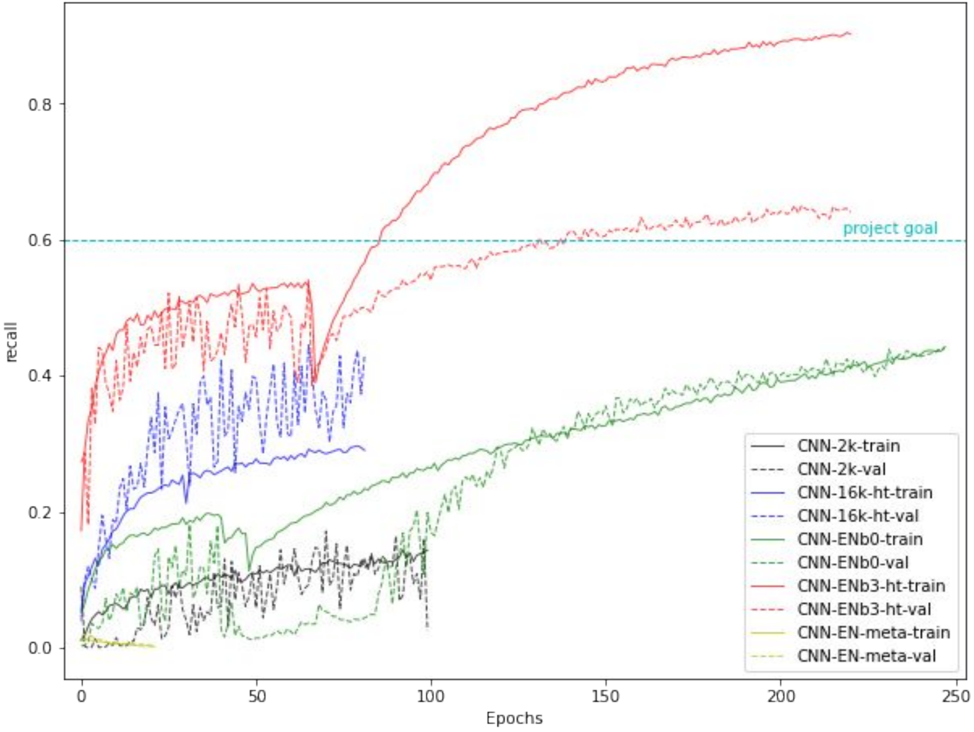


Model abbreviations:

- *CNN-2k*: small dataset
- *CNN-16k-ht*: big dataset Optuna-tuned
- *CNN-ENb0*: EfficientNetB0
- *CNN-ENb3-ht*: EfficientNetB3 with Optuna hyperparameter-tuning
- *CNN-EN-meta*: EfficientNet with metadata

Note: these are categorical accuracies (based on model output probabilities)

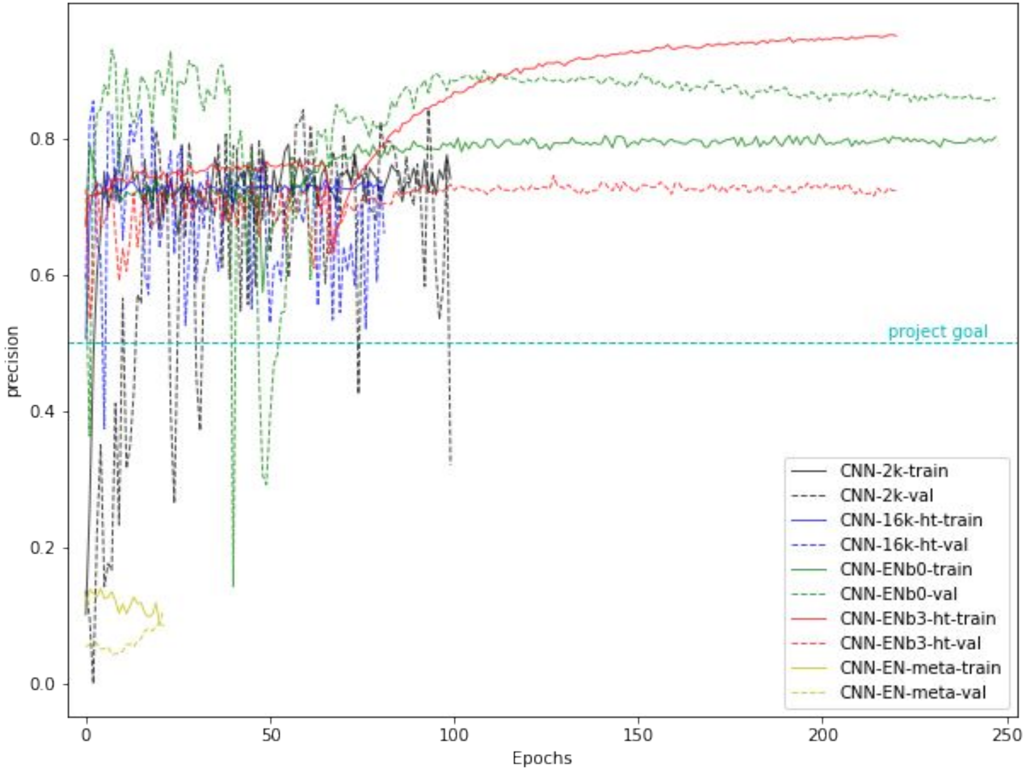
Recall: top CNN models



Model abbreviations:

- *CNN-2k*: small dataset
- *CNN-16k-ht*: big dataset Optuna-tuned
- *CNN-ENb0*: EfficientNetB0
- *CNN-ENb3-ht*: EfficientNetB3 with Optuna hyperparameter-tuning
- *CNN-EN-meta*: EfficientNet with metadata

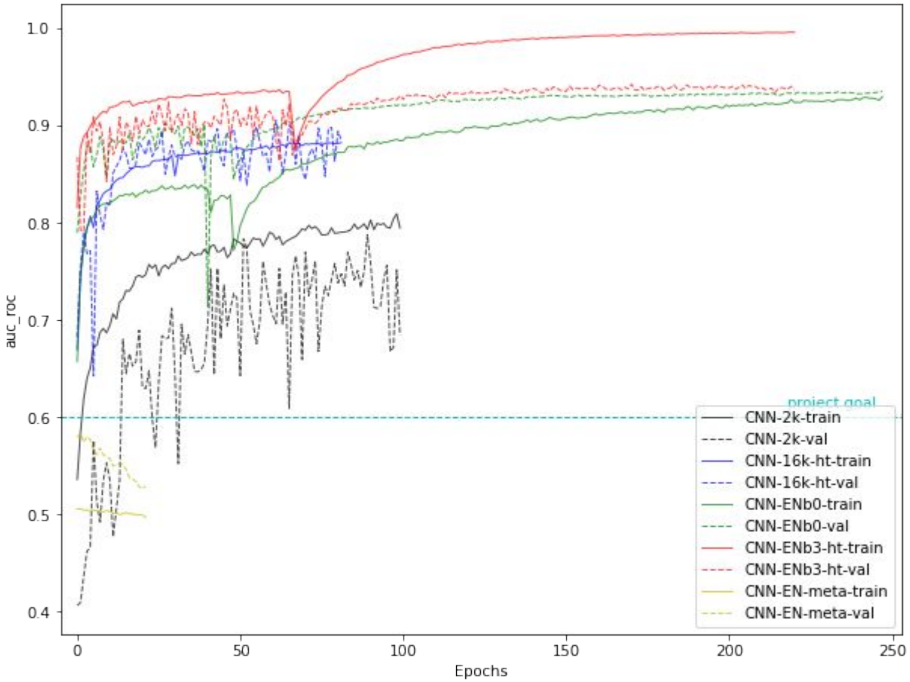
Precision: top CNN models



Model abbreviations:

- *CNN-2k*: small dataset
- *CNN-16k-ht*: big dataset Optuna-tuned
- *CNN-ENb0*: EfficientNetB0
- *CNN-ENb3-ht*: EfficientNetB3 with Optuna hyperparameter-tuning
- *CNN-EN-meta*: EfficientNet with metadata

AUC-ROC: top CNN models



Model abbreviations:

- *CNN-2k*: small dataset
- *CNN-16k-ht*: big dataset Optuna-tuned
- *CNN-ENb0*: EfficientNetB0
- *CNN-ENb3-ht*: EfficientNetB3 with Optuna hyperparameter-tuning
- *CNN-EN-meta*: EfficientNet with metadata

Metadata Clustering

K-Means vs K-Prototypes

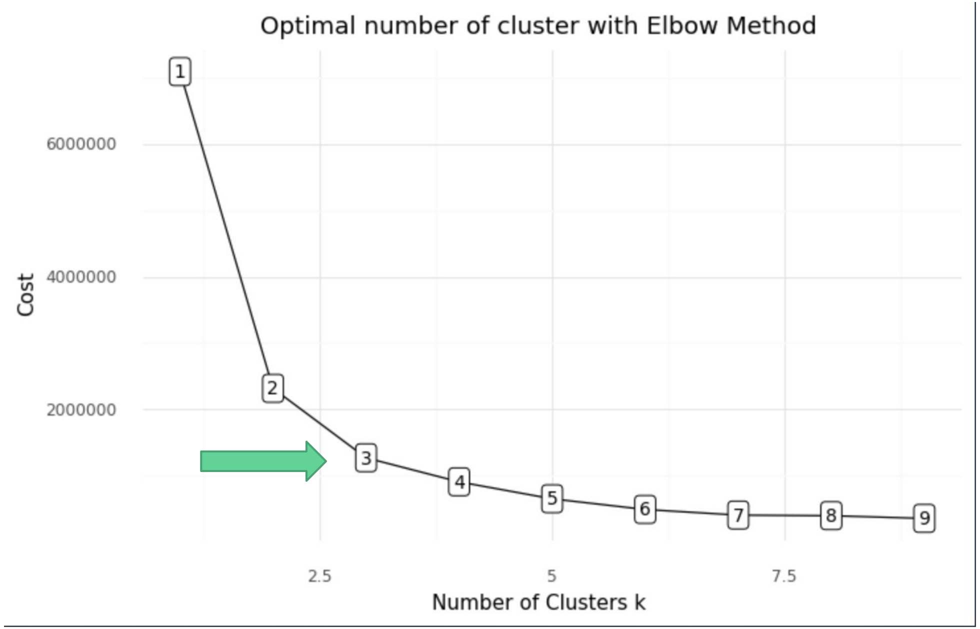
- Our metadata contains a mix of numerical and categorical data
- The standard K-means algorithm isn't directly applicable to categorical data
  - Categorical data is discrete
  - A Euclidean distance isn't really meaningful
- k-prototypes is a combination of k-means and k-modes.
  - Can handle both numerical and categorical features simultaneously

	image	age_approx	anatom_site_general	lesion_id	sex
0	ISIC_0000000	55.0	anterior torso	NaN	female
1	ISIC_0000001	30.0	anterior torso	NaN	female
2	ISIC_0000002	60.0	upper extremity	NaN	female
3	ISIC_0000003	30.0	upper extremity	NaN	male
4	ISIC_0000004	80.0	posterior torso	NaN	male
...	...	...	...	...	...
25326	ISIC_0073247	85.0	head/neck	BCN_0003925	female
25327	ISIC_0073248	65.0	anterior torso	BCN_0001819	male
25328	ISIC_0073249	70.0	lower extremity	BCN_0001085	male
25329	ISIC_0073251	55.0	palms/soles	BCN_0002083	female
25330	ISIC_0073254	50.0	upper extremity	BCN_0001079	male

25331 rows × 5 columns

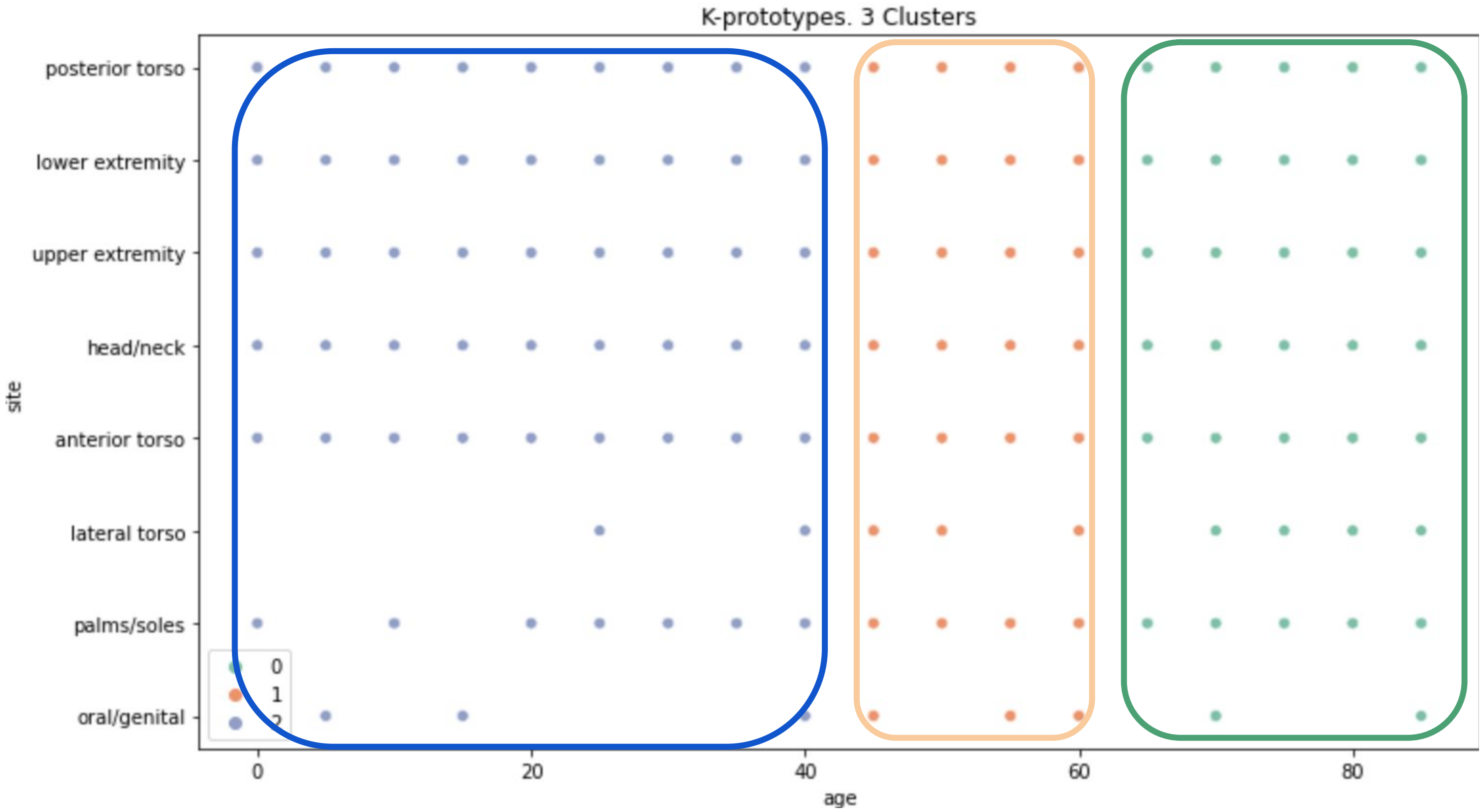
K-prototypes: Choosing K

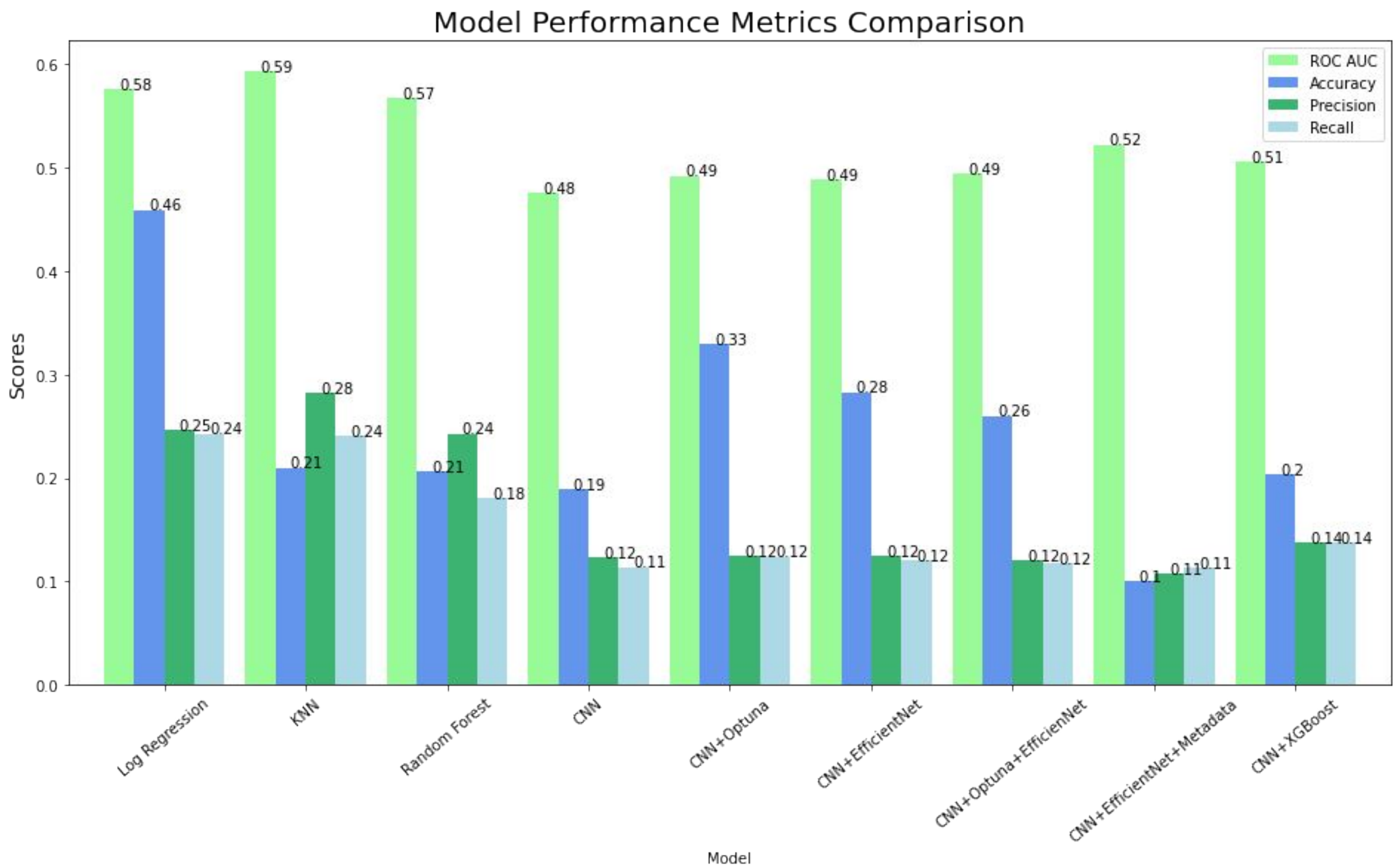
Elbow method [1:10]



K-prototypes

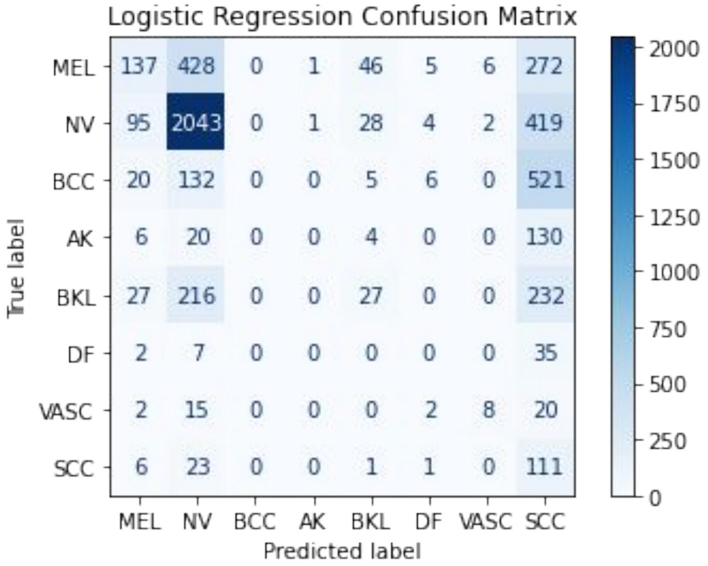
K=3



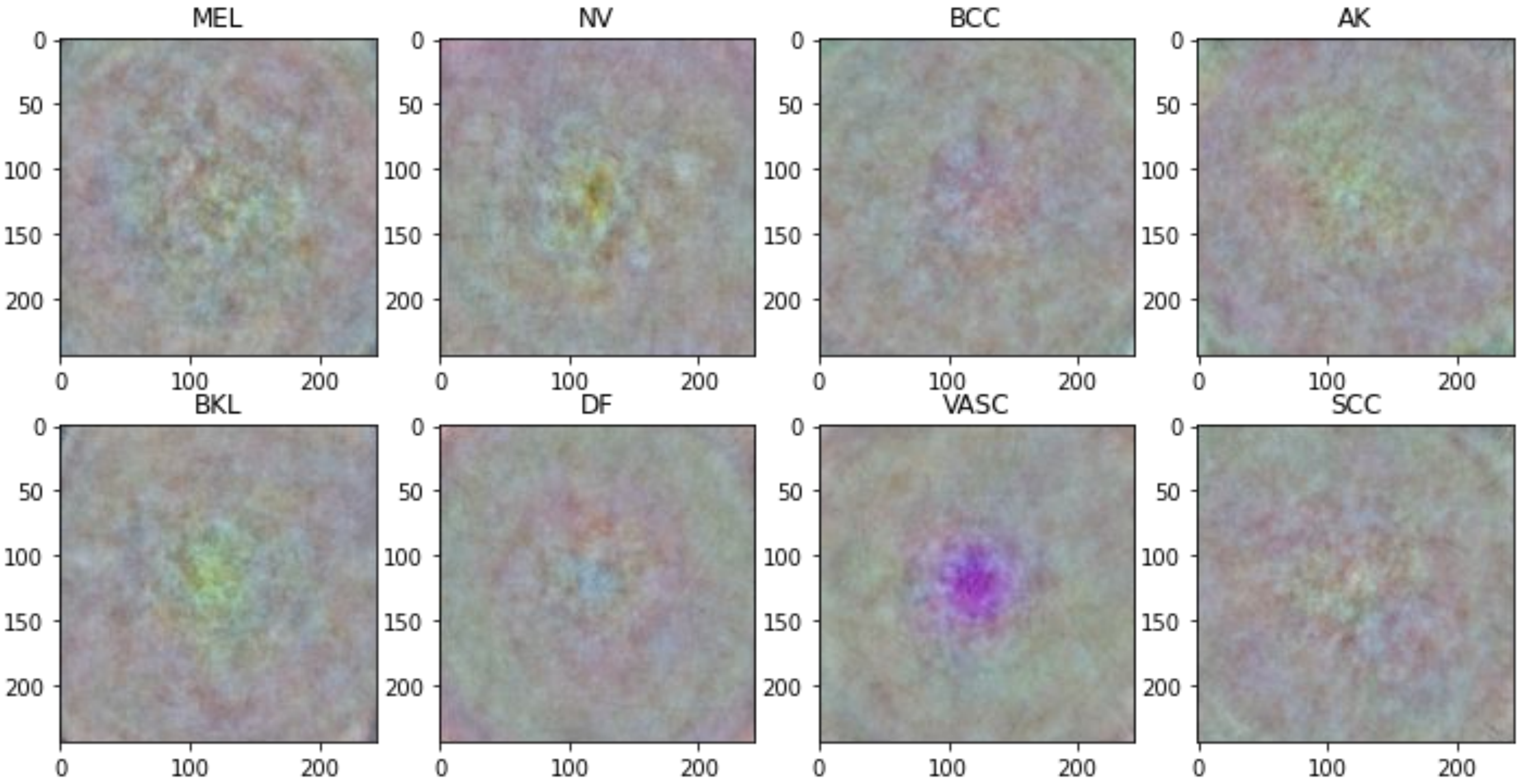

Model Performance Metrics Comparison
ROC AUC
Accuracy
Precision
Recall
Scores
Model
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.58
0.46
0.25
0.24
0.59
0.21
0.28
0.24
0.57
0.21
0.24
0.18
0.48
0.19
0.12
0.11
0.49
0.33
0.12
0.12
0.49
0.28
0.12
0.12
0.49
0.26
0.12
0.12
0.52
0.1
0.11
0.11
0.51
0.2
0.14
0.14
Log Regression
KNN
Random Forest
CNN
CNN+Optuna
CNN+EfficientNet
CNN+Optuna+EfficienNet
CNN+EfficientNet+Metadata
CNN+XGBoost

Model Conclusions

- Overall, model performance was poor
- None of our models met performance thresholds
- All models had difficulty with NV vs. MEL classification
    - CNN and Logistic Regression had best recall score for melanoma
    - CNN + tuning or transfer learning improved performance more for benign classes
- Performance not sufficient for clinical use



Logistic regression weights for each skin class



Strengths

- Real-world dataset representing real variability
- Image Augmentation implemented for class balance
- Cropping method implemented
- Optimization utilized across models
    - Optimized K in KNN and n_estimators in random forest
    - Elbow method for selecting number of clusters in K-prototypes
    - Tested performance across different EfficientNet versions

Constraints and Limitations

- Smaller training sets due to computational limitations
- Reduced generalizability for different races
- Future directions:
  - Principal component analysis
  - Further image augmentation
  - Assessing impact of presence of rulers, marks, hair, etc for certain classes

References

[1] International Skin Imaging Collaboration (ISIC) on Kaggle, 2020: https://www.kaggle.com/competitions/siim-isic-melanoma-classification/overview

[2] Cancer.net, February 2022. Melanoma: Statistics. https://www.cancer.net/cancer-types/melanoma/statistics#:~:text=It%20is%20estimated%20that%207%2C650,people%20worldwide%20died%20from%20melanoma.

[3] ISIC Archive : https://www.isic-archive.com/

[4] Tschandl P., Rosendahl C. & Kittler H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci. Data 5, 180161 doi.10.1038/sdata.2018.161 (2018)

[5] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, Allan Halpern: "Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)", 2017; arXiv: 1710.05006.

[6] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Allan C. Halpern, Susana Puig, Josep Malvehy: "BCN20000: Dermoscopic Lesions in the Wild", 2019; arXiv:1908.02288.

[7] Haenssle, H. A., et al. (2018). Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. Annals of oncology : official journal of the European Society for Medical Oncology, 29(8), 1836–1842. https://doi.org/10.1093/annonc/mdy166

[8] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta,and Masanori Koyama. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. In KDD. https://optuna.org/

[9] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv. https://doi.org/10.48550/arXiv.1905.11946

[10] dmlc XGBoost read-the-docs website (2022). https://xgboost.readthedocs.io/en/stable/index.html

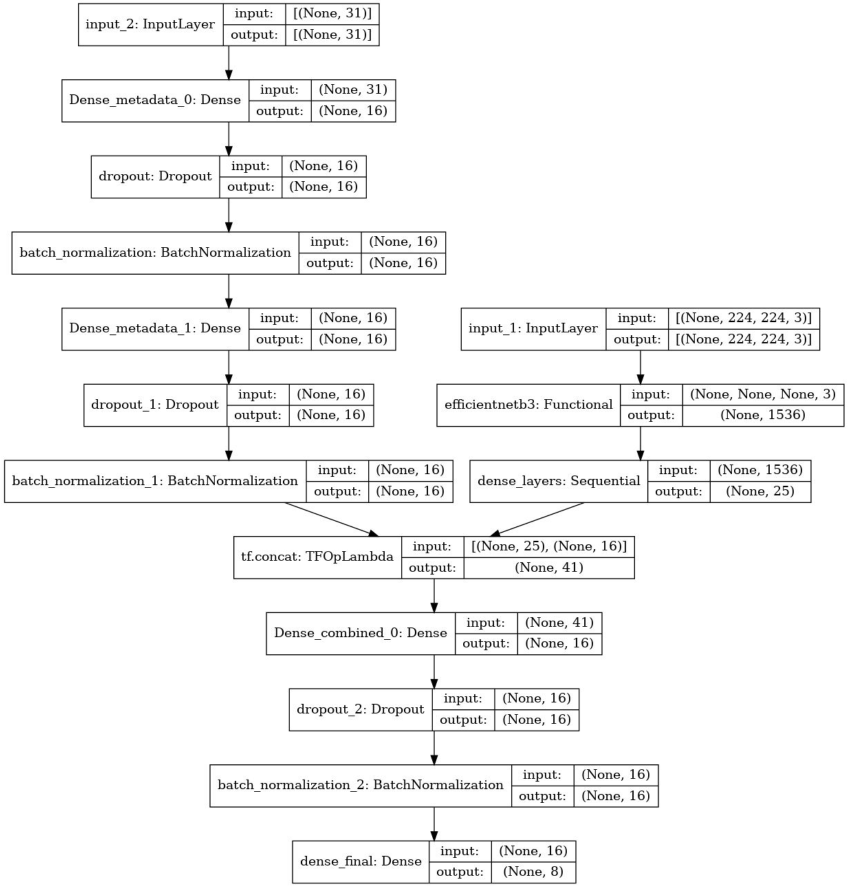
[11] Ren, X., Guo, H., Li, S., Wang, S., Li, J. (2017). A Novel Image Classification Method with CNN-XGBoost Model. In: Kraetzer, C., Shi, YQ., Dittmann, J., Kim, H. (eds) Digital Forensics and Watermarking. IWDW 2017. Lecture Notes in Computer Science(), vol 10431. Springer, Cham. https://doi.org/10.1007/978-3-319-64185-0_28

[12] Chollet, F (2020) Keras, Transfer learning & fine-tuning. https://keras.io/guides/transfer_learning/

Appendix

CNN + Metadata structure

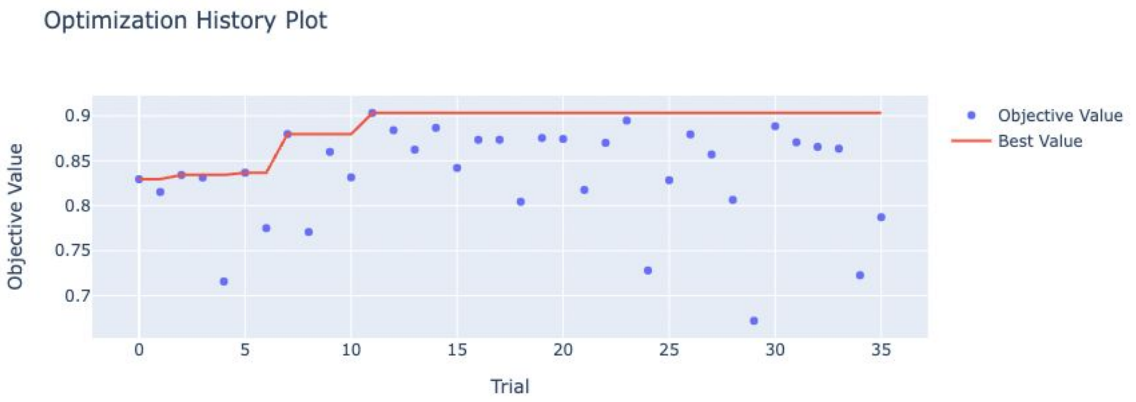
Example structure of model using metadata input (normal dense layers; left) and image data input (CNN; right), which gets concatenated and put through a dense layer before classification.



Optuna Hyperparameter Tuning

Hyperparameter tuning of CNN structure

1. *A* sets of: (options: 2, 3)
   - Conv2D with:
     - *B* filters (options: 5 to 16)
     - *C* kernel size (options: 3 to 5)
     - *D* setting of padding/no padding (options: True/False)
     - activation: relu
   - MaxPooling
     - *E* pool size (options: 2, 3)
     - *F* stride (options: 2, 3)
   - Dropout: 0.1
2. Flatten
3. *G* layers of Dense (options: 0 to 3)
   - *H* units (options: 5 to 20)
   - Activation: relu
   - Dropout: 0.1
4. One output layer of Dense
   - output units: 8
   - Activation: softmax



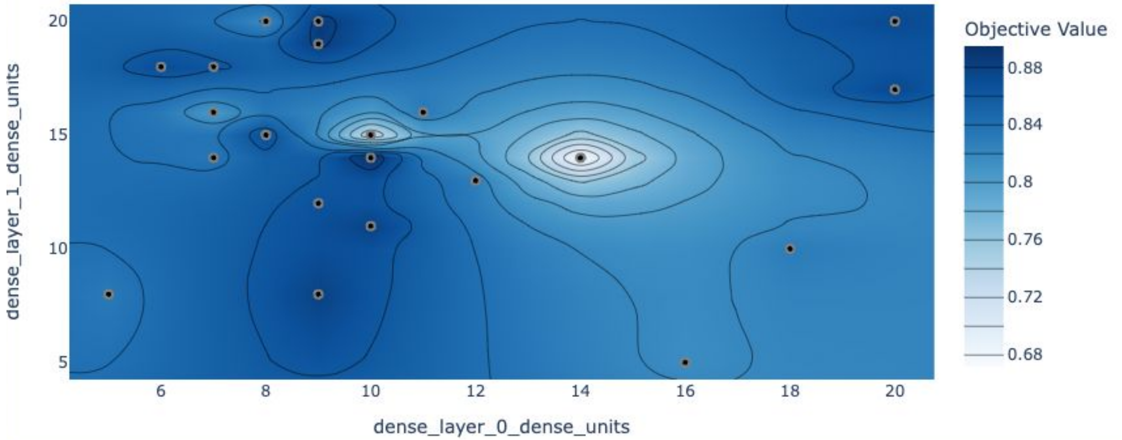
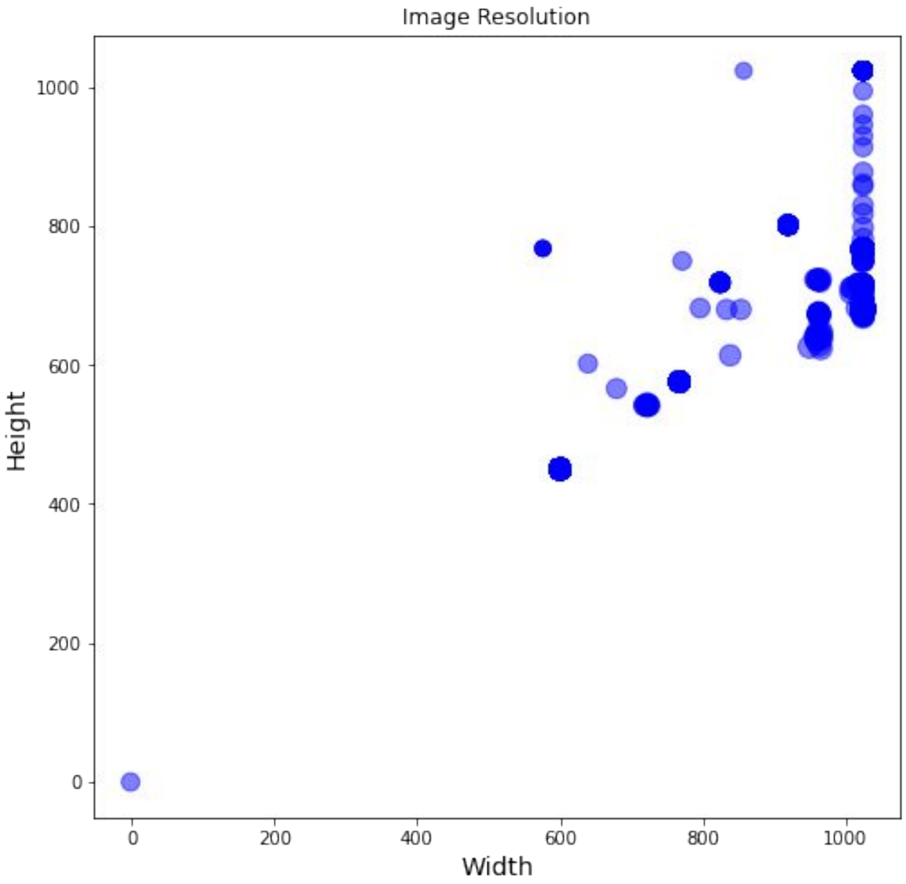


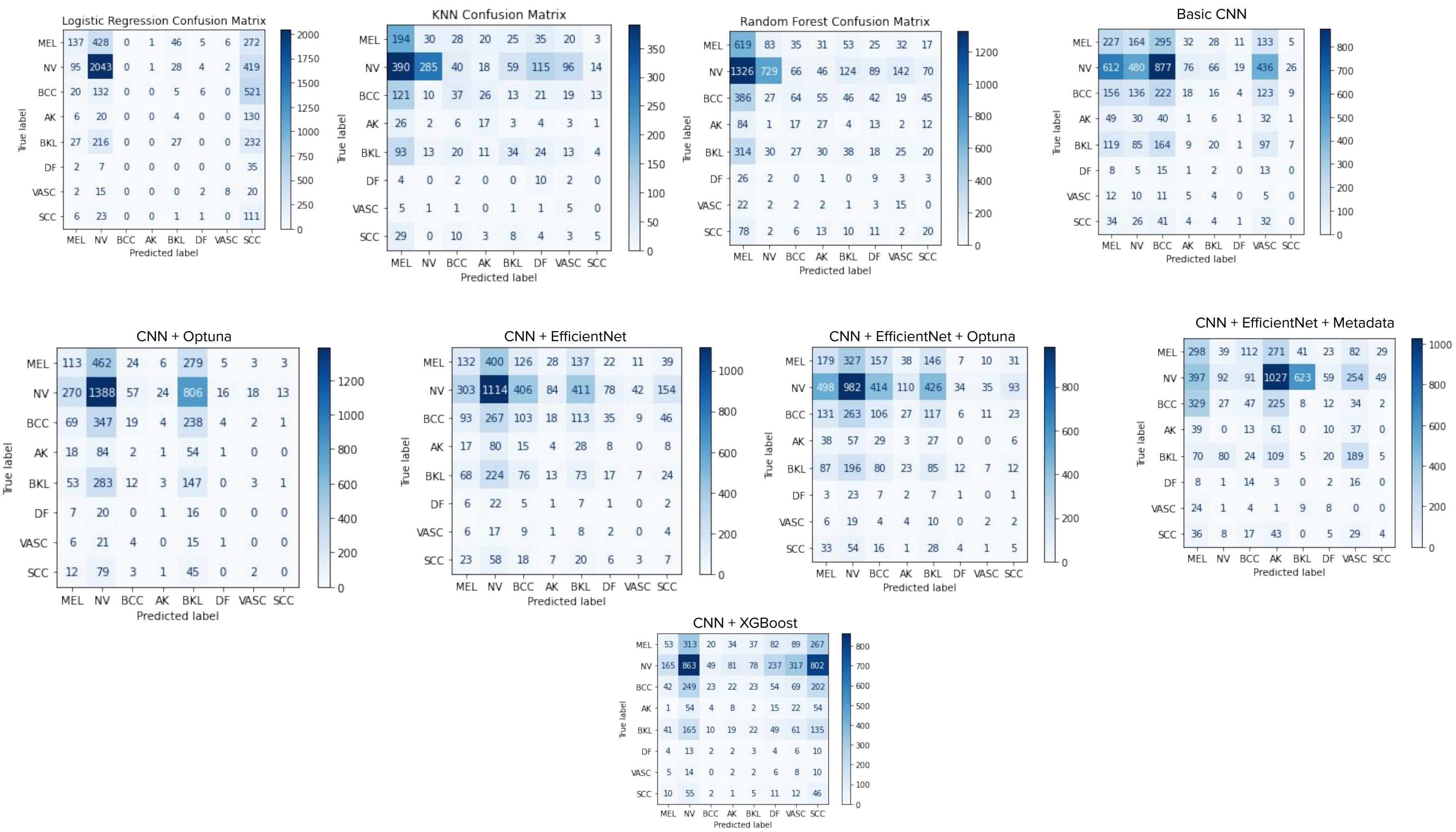
Image size distributions

Variable, but two main groups:

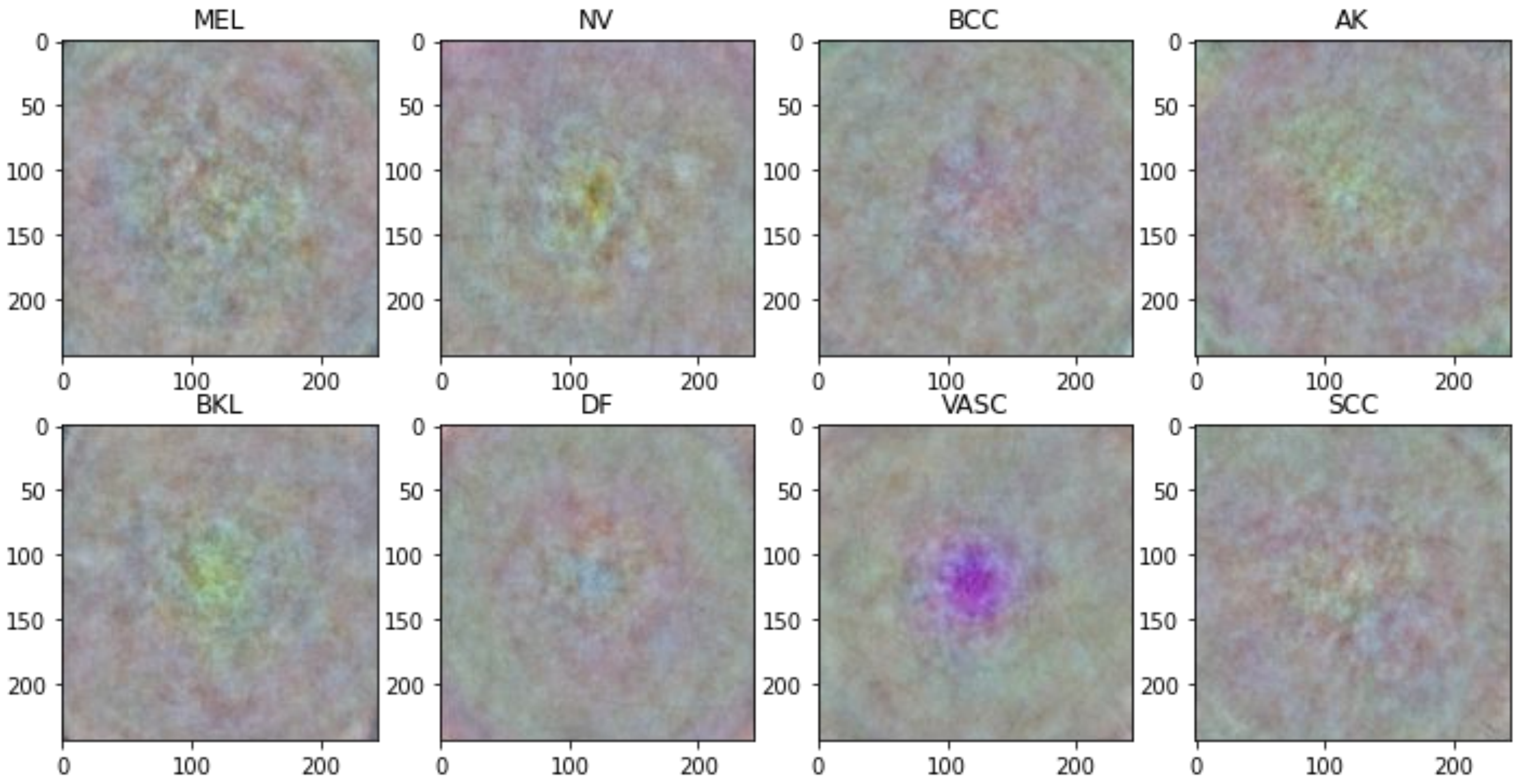
10,014 images of 600 x 450

12,410 images of 1024 x 1024




Logistic Regression Confusion Matrix
True label
MEL 137 428 0 1 46 5 6 272
NV 95 2043 0 1 28 4 2 419
BCC 20 132 0 0 5 6 0 521
AK 6 20 0 0 4 0 0 130
BKL 27 216 0 0 27 0 0 232
DF 2 7 0 0 0 0 0 35
VASC 2 15 0 0 0 2 8 20
SCC 6 23 0 0 1 1 0 111
MEL NV BCC AK BKL DF VASC SCC
Predicted label
2000 1750 1500 1250 1000 750 500 250 0
KNN Confusion Matrix
True label
MEL 194 30 28 20 25 35 20 3
NV 390 285 40 18 59 115 96 14
BCC 121 10 37 26 13 21 19 13
AK 26 2 6 17 3 4 3 1
BKL 93 13 20 11 34 24 13 4
DF 4 0 2 0 0 10 2 0
VASC 5 1 1 0 1 1 5 0
SCC 29 0 10 3 8 4 3 5
MEL NV BCC AK BKL DF VASC SCC
Predicted label
350 300 250 200 150 100 50 0
Random Forest Confusion Matrix
True label
MEL 619 83 35 31 53 25 32 17
NV 1326 729 66 46 124 89 142 70
BCC 386 27 64 55 46 42 19 45
AK 84 1 17 27 4 13 2 12
BKL 314 30 27 30 38 18 25 20
DF 26 2 0 1 0 9 3 3
VASC 22 2 2 2 1 3 15 0
SCC 78 2 6 13 10 11 2 20
MEL NV BCC AK BKL DF VASC SCC
Predicted label
1200 1000 800 600 400 200 0
Basic CNN
True label
MEL 227 164 295 32 28 11 133 5
NV 612 480 877 76 66 19 436 26
BCC 156 136 222 18 16 4 123 9
AK 49 30 40 1 6 1 32 1
BKL 119 85 164 9 20 1 97 7
DF 8 5 15 1 2 0 13 0
VASC 12 10 11 5 4 0 5 0
SCC 34 26 41 4 4 1 32 0
MEL NV BCC AK BKL DF VASC SCC
Predicted label
800 700 600 500 400 300 200 100 0
CNN + Optuna
True label
MEL 113 462 24 6 279 5 3 3
NV 270 1388 57 24 806 16 18 13
BCC 69 347 19 4 238 4 2 1
AK 18 84 2 1 54 1 0 0
BKL 53 283 12 3 147 0 3 1
DF 7 20 0 1 16 0 0 0
VASC 6 21 4 0 15 1 0 0
SCC 12 79 3 1 45 0 2 0
MEL NV BCC AK BKL DF VASC SCC
Predicted label
1200 1000 800 600 400 200 0
CNN + EfficientNet
True label
MEL 132 400 126 28 137 22 11 39
NV 303 1114 406 84 411 78 42 154
BCC 93 267 103 18 113 35 9 46
AK 17 80 15 4 28 8 0 8
BKL 68 224 76 13 73 17 7 24
DF 6 22 5 1 7 1 0 2
VASC 6 17 9 1 8 2 0 4
SCC 23 58 18 7 20 6 3 7
MEL NV BCC AK BKL DF VASC SCC
Predicted label
1000 800 600 400 200 0
CNN + EfficientNet + Optuna
True label
MEL 179 327 157 38 146 7 10 31
NV 498 982 414 110 426 34 35 93
BCC 131 263 106 27 117 6 11 23
AK 38 57 29 3 27 0 0 6
BKL 87 196 80 23 85 12 7 12
DF 3 23 7 2 7 1 0 1
VASC 6 19 4 4 10 0 2 2
SCC 33 54 16 1 28 4 1 5
MEL NV BCC AK BKL DF VASC SCC
Predicted label
800 600 400 200 0
CNN + EfficientNet + Metadata
True label
MEL 298 39 112 271 41 23 82 29
NV 397 92 91 1027 623 59 254 49
BCC 329 27 47 225 8 12 34 2
AK 39 0 13 61 0 10 37 0
BKL 70 80 24 109 5 20 189 5
DF 8 1 14 3 0 2 16 0
VASC 24 1 4 1 9 8 0 0
SCC 36 8 17 43 0 5 29 4
MEL NV BCC AK BKL DF VASC SCC
Predicted label
1000 800 600 400 200 0
CNN + XGBoost
True label
MEL 53 313 20 34 37 82 89 267
NV 165 863 49 81 78 237 317 802
BCC 42 249 23 22 23 54 69 202
AK 1 54 4 8 2 15 22 54
BKL 41 165 10 19 22 49 61 135
DF 4 13 2 2 3 4 6 10
VASC 5 14 0 2 2 6 8 10
SCC 10 55 2 1 5 11 12 46
MEL NV BCC AK BKL DF VASC SCC
Predicted label
800 700 600 500 400 300 200 100 0

Visualizing logistic regression weights for each skin class



Logistic Regression Classification Report

	precision	recall	f1-score	support
MEL	0.464	0.153	0.230	895
NV	0.708	0.788	0.746	2592
BCC	0.000	0.000	0.000	684
AK	0.000	0.000	0.000	160
BKL	0.243	0.054	0.088	502
DF	0.000	0.000	0.000	44
VASC	0.500	0.170	0.254	47
SCC	0.064	0.782	0.118	142
accuracy			0.459	5066
macro avg	0.247	0.243	0.180	5066
weighted avg	0.475	0.459	0.437	5066

Accuracy: 0.4591
ROC AUC: 0.5760

KNN Classification Report

	precision	recall	f1-score	support
MEL	0.540	0.076	0.133	355
NV	0.836	0.280	0.420	1017
BCC	0.257	0.142	0.183	260
AK	0.179	0.274	0.217	62
BKL	0.238	0.160	0.192	212
DF	0.047	0.556	0.086	18
VASC	0.031	0.357	0.057	14
SCC	0.125	0.081	0.098	62
micro avg	0.354	0.210	0.263	2000
macro avg	0.282	0.241	0.173	2000
weighted avg	0.590	0.210	0.292	2000
samples avg	0.210	0.210	0.210	2000

Accuracy: 0.21
ROC AUC: 0.5933327707205982

Random Forest Classification Report

	precision	recall	f1-score	support
MEL	0.336	0.164	0.221	895
NV	0.832	0.281	0.420	2592
BCC	0.295	0.094	0.142	684
AK	0.132	0.169	0.148	160
BKL	0.138	0.076	0.098	502
DF	0.043	0.205	0.071	44
VASC	0.062	0.319	0.105	47
SCC	0.107	0.141	0.122	142
micro avg	0.396	0.207	0.272	5066
macro avg	0.243	0.181	0.166	5066
weighted avg	0.547	0.207	0.293	5066
samples avg	0.207	0.207	0.207	5066

Accuracy: 0.20706671930517173
ROC AUC: 0.5672968987988957

CNN Classification Report

	precision	recall	f1-score	support
MEL	0.187	0.254	0.215	895
NV	0.513	0.185	0.272	2592
BCC	0.133	0.325	0.189	684
AK	0.007	0.006	0.007	160
BKL	0.137	0.040	0.062	502
DF	0.000	0.000	0.000	44
VASC	0.006	0.106	0.011	47
SCC	0.000	0.000	0.000	142
accuracy			0.189	5066
macro avg	0.123	0.114	0.094	5066
weighted avg	0.327	0.189	0.209	5066

Accuracy: 0.1885
ROC AUC: 0.4760

CNN + Optuna Classification Report

	precision	recall	f1-score	support
MEL	0.206	0.126	0.157	895
NV	0.517	0.535	0.526	2592
BCC	0.157	0.028	0.047	684
AK	0.025	0.006	0.010	160
BKL	0.092	0.293	0.140	502
DF	0.000	0.000	0.000	44
VASC	0.000	0.000	0.000	47
SCC	0.000	0.000	0.000	142
accuracy			0.329	5066
macro avg	0.125	0.124	0.110	5066
weighted avg	0.332	0.329	0.317	5066

Accuracy: 0.3293
ROC AUC: 0.4918

CNN + EfficientNet Classification Report

	precision	recall	f1-score	support
MEL	0.204	0.147	0.171	895
NV	0.511	0.430	0.467	2592
BCC	0.136	0.151	0.143	684
AK	0.026	0.025	0.025	160
BKL	0.092	0.145	0.112	502
DF	0.006	0.023	0.009	44
VASC	0.000	0.000	0.000	47
SCC	0.025	0.049	0.033	142
accuracy			0.283	5066
macro avg	0.125	0.121	0.120	5066
weighted avg	0.326	0.283	0.301	5066

Accuracy: 0.2831
ROC AUC: 0.4891

CNN+Optuna+EfficientNet Classification Report

	precision	recall	f1-score	support
MEL	0.171	0.187	0.179	895
NV	0.500	0.371	0.426	2592
BCC	0.118	0.140	0.128	684
AK	0.024	0.031	0.027	160
BKL	0.092	0.155	0.116	502
DF	0.000	0.000	0.000	44
VASC	0.000	0.000	0.000	47
SCC	0.052	0.063	0.057	142
accuracy			0.260	5066
macro avg	0.120	0.118	0.117	5066
weighted avg	0.314	0.260	0.281	5066

Accuracy: 0.2598
ROC AUC: 0.4937