# CAD Project I: Classical Machine Learning

KAOUTHER MOUHEB

RACHIKA ELHASSNA HAMADACHE

MAIA 2021-2023

# Outline

1. Proposal Analysis

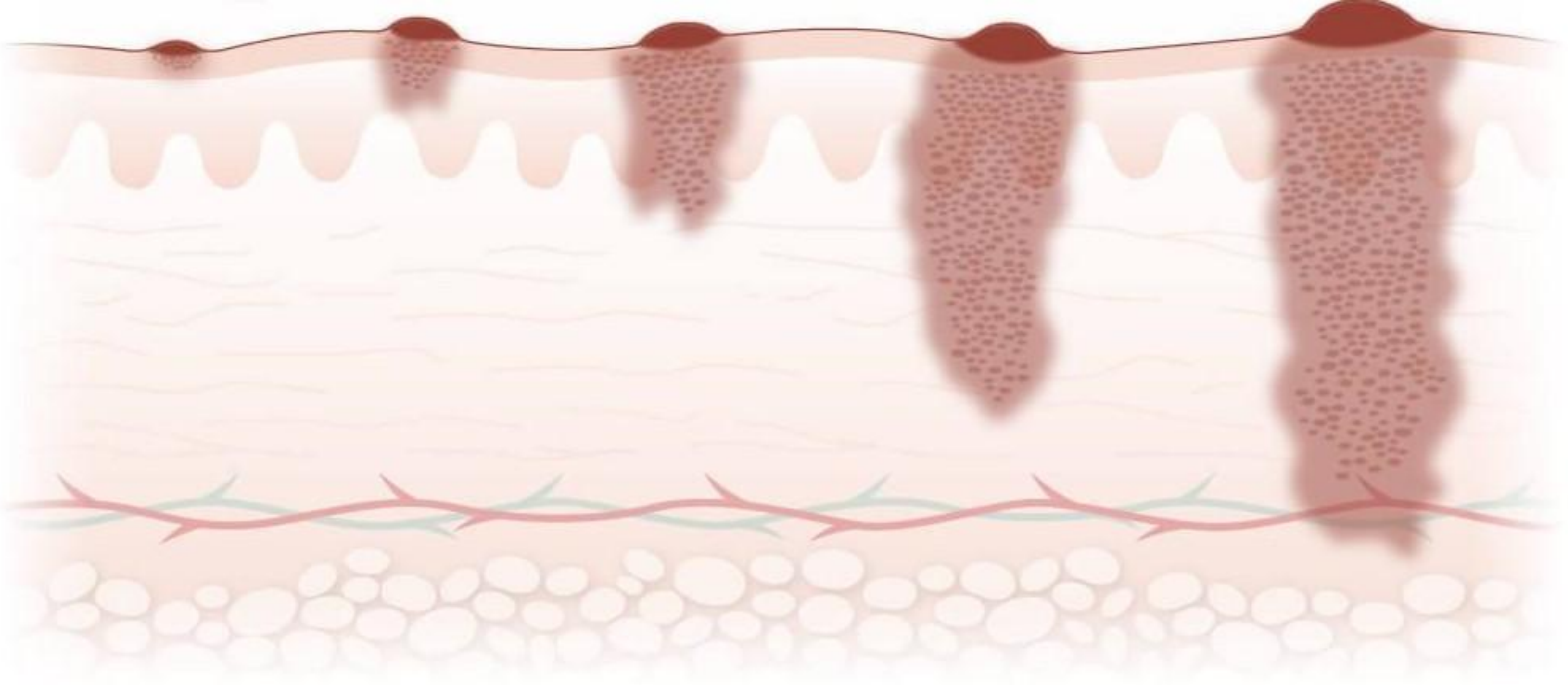
2. Implementation and Design

    1. Preprocessing
    2. Feature Extraction
    3. Sampling
    4. Parameter Fine-tuning
    5. Feature Selection
    6. Model Selection

3. Experimental Results and Evaluation

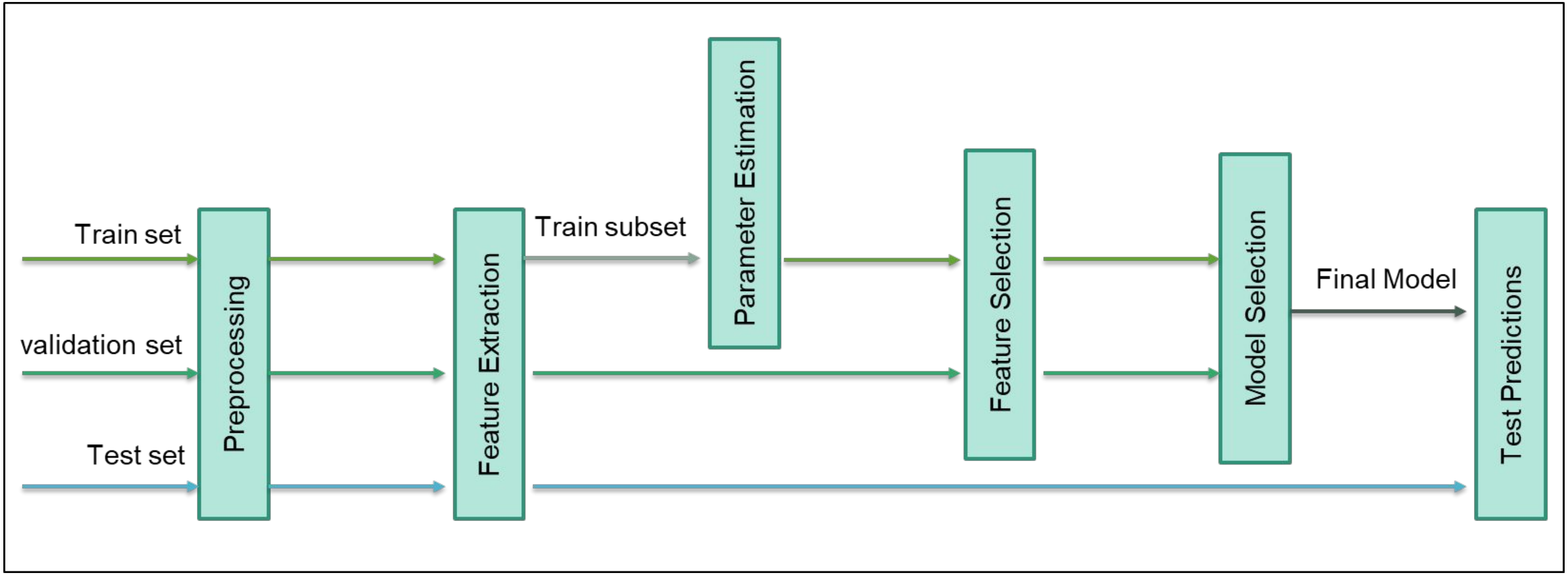
4. Bag Of Visual Words

5. Conclusion

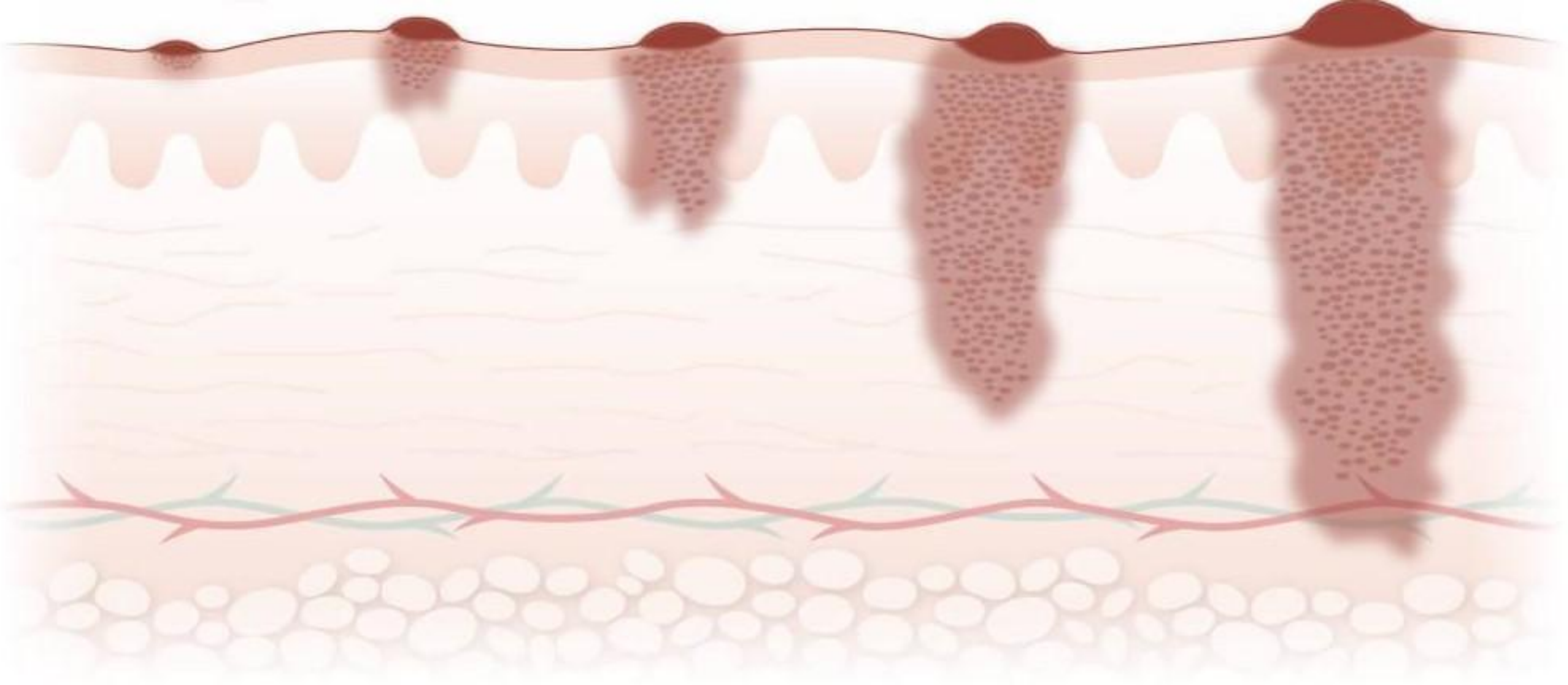


# Proposal Analysis

# Proposed Pipeline




Proposed Pipeline



# Preprocessing

# Hair removal

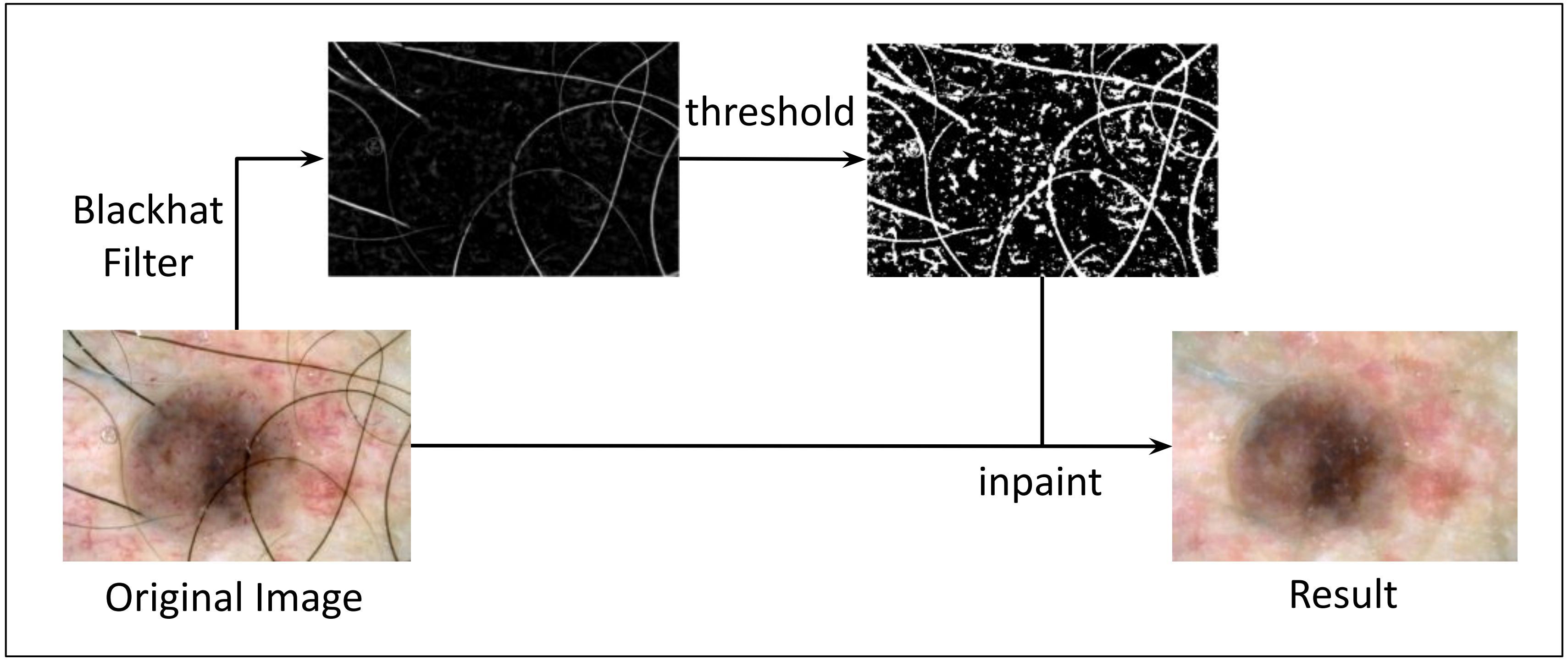
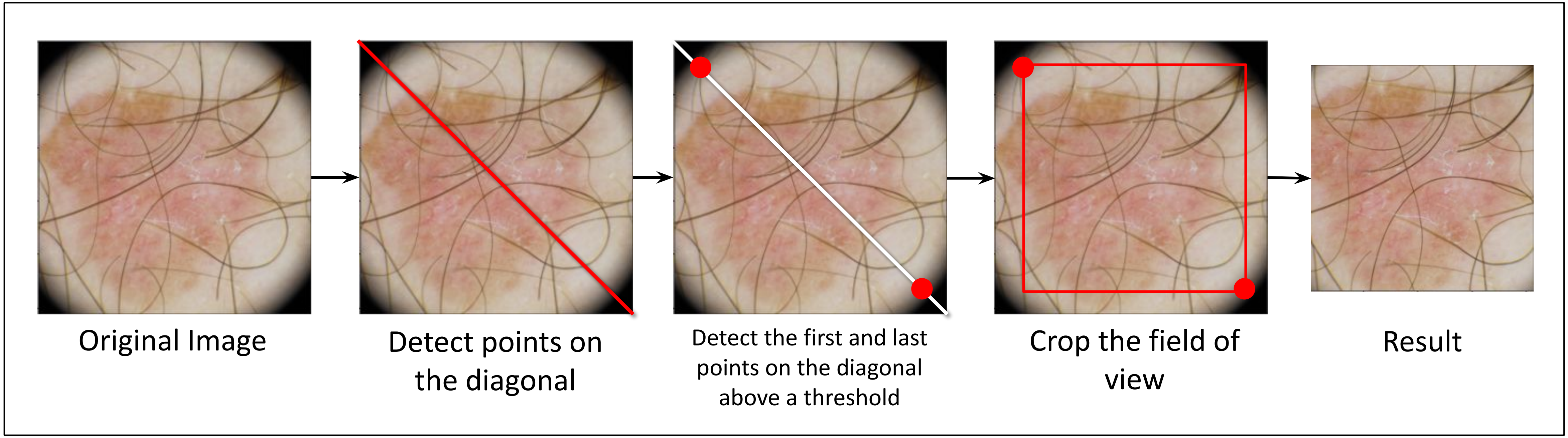



Figure 2: Hair removal pipeline

# Vignette removal




Vignette removal pipeline

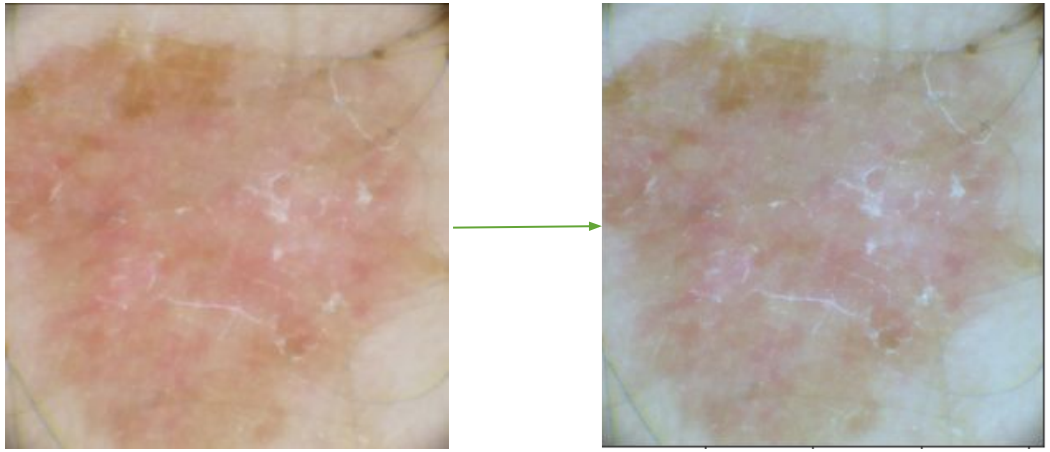
# Color Constancy

A *color compensation* technique to reduce the influence of the acquisition setup on the color features extracted from the images. It uses the Minkowski norm to estimate the color of the illuminant.

The illuminant *e* is obtained as follows:

$$\left(\frac{\int (I_c(\mathbf{x}))^p \mathrm{d}\mathbf{x}}{\int \mathrm{d}\mathbf{x}}\right)^{1/p} = ke_c$$

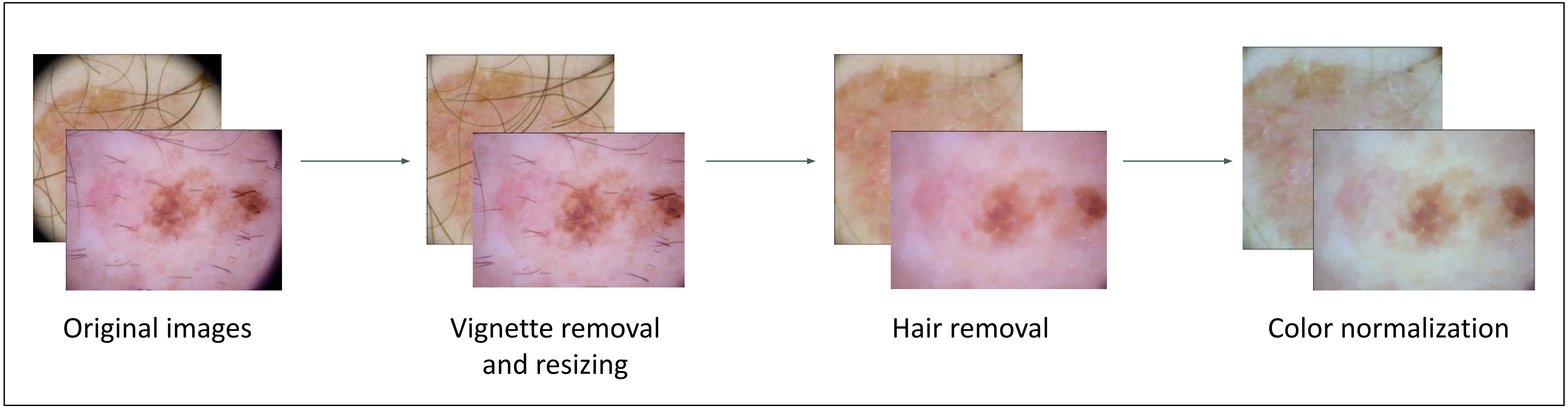
Where *Ic* denotes the c$^{th}$ component of the image, k a normalization factor, p the degree of the norm.



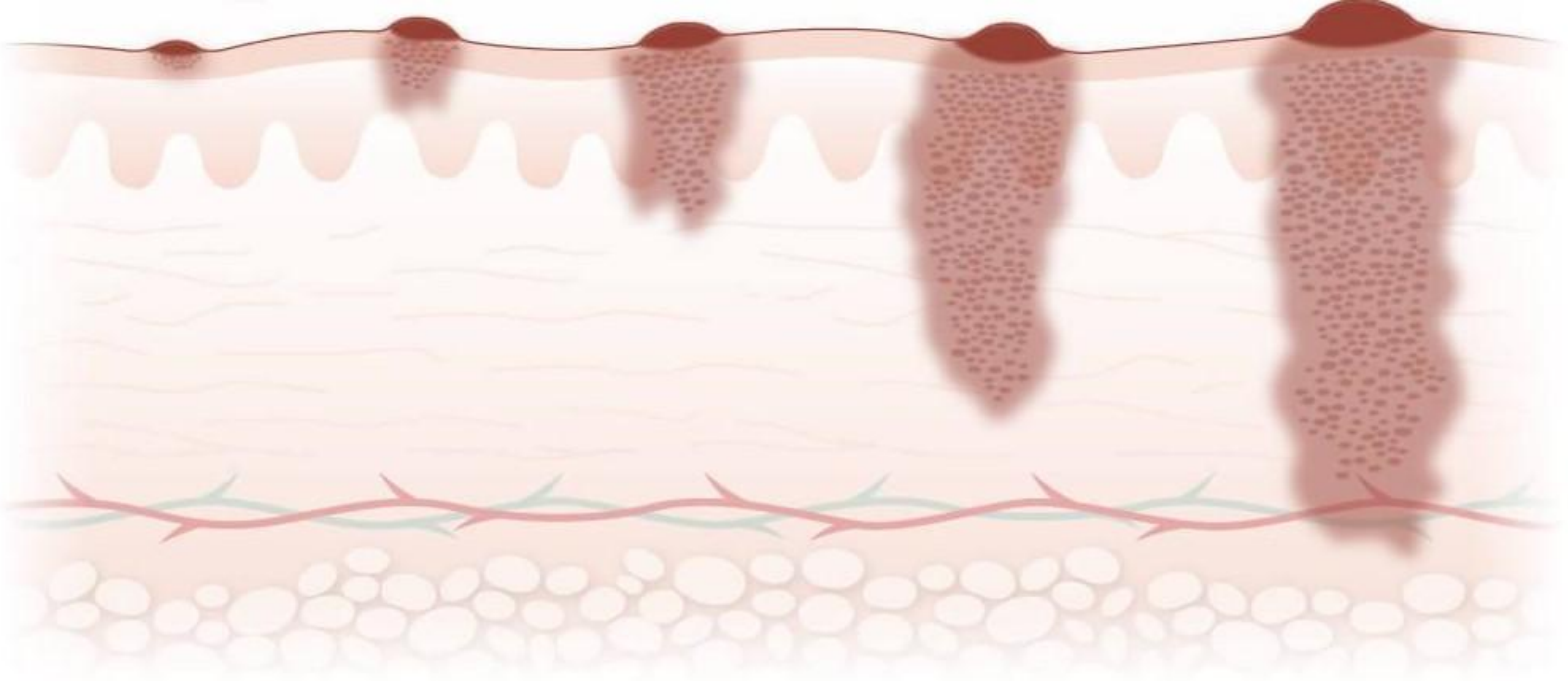
Before Normalization After Normalization

Color Normalization Example

# Full Preprocessing Pipeline




Preprocessing pipeline



# Design and Implementation

# Feature Extraction

# Color features

1. Color variegation : quantified by the normalized standard deviation of the red, green and blue components of the image.
2. Color moments : 4 measures characterizing the color distribution of the images : mean, standard deviation, skewness and variance.
3. Color histograms : histogram of each channel of the 3 color spaces : RGB, HSV and Lab

# Texture features

1. LBP : Local binary patterns with P=16 and R=2.
2. Haralick : texture features based on the GLCM matrix and computing 13 measures.
3. GLCM : 5 main properties computed for 4 different angles with unit step : correlation, homogeneity, contrast, energy and dissimilarity

# Parameter Fine-tuning

# Parameter Estimation

- For challenge 1, a subset of 4000 images (2000 per class) was sampled to perform fine-tuning.
- All features are used for this step.
- GridSearchCV was used with 5 folds to estimate the Following parameters:

**Random Forest and Extra Trees:**

- Maximum depth
- Number of Estimators
- Criterion

**XGBoost:**

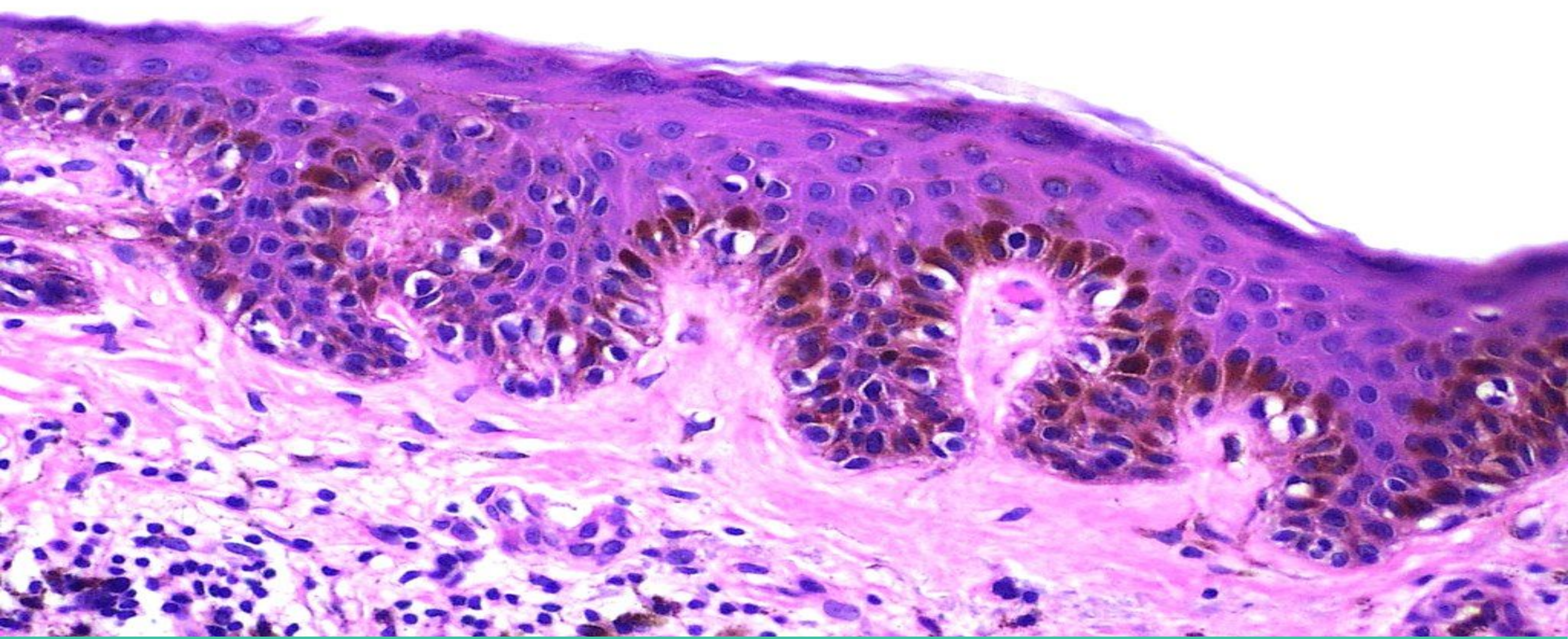
- Maximum depth
- Number of Estimators
- Learning rate

**SVM:**

- Penalization parameter C.

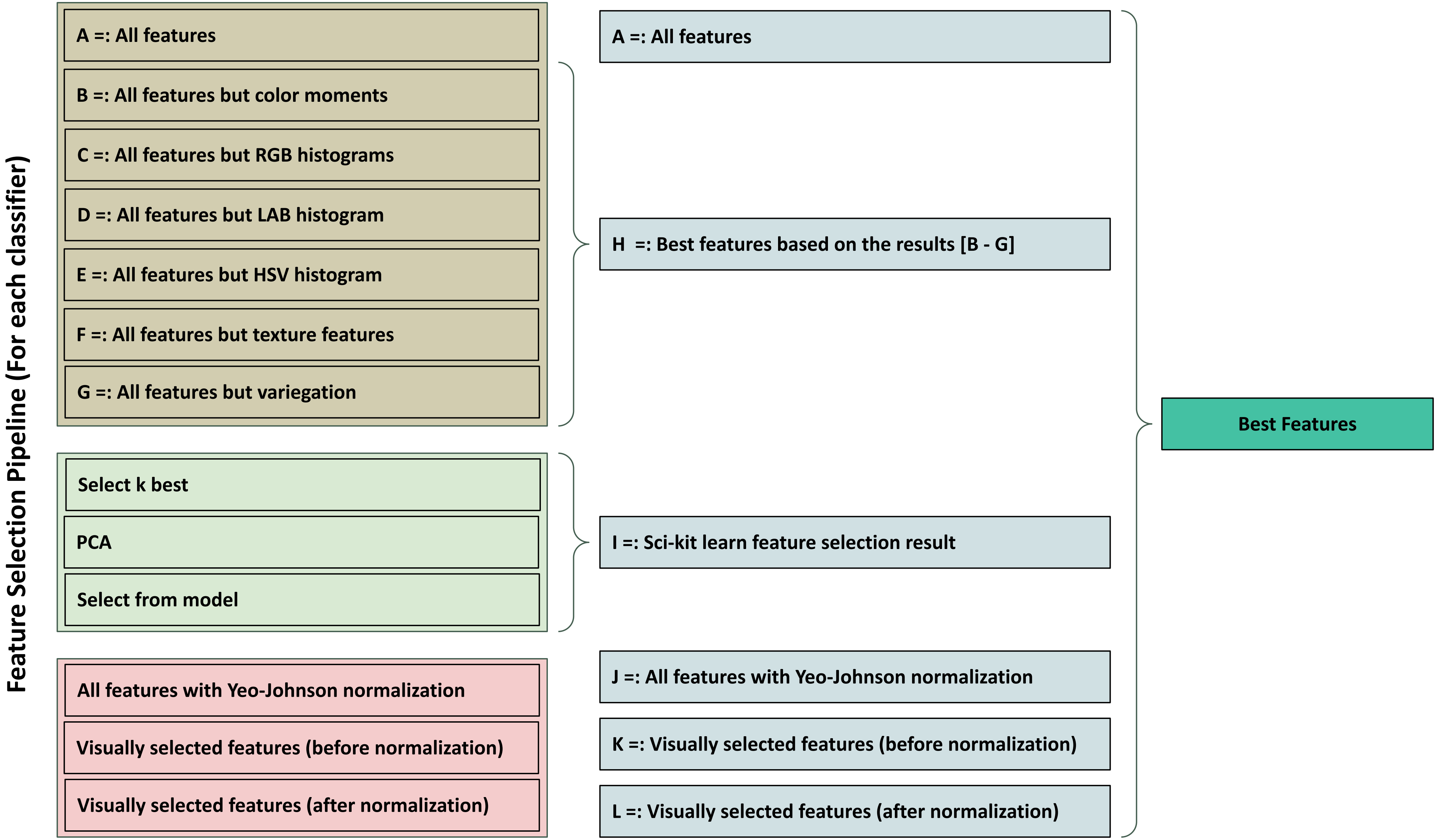
**K-NN:**

- The number of neighbors k.



# Challenge 1: Nevus Versus Others

# Feature Selection


Feature Selection Pipeline (For each classifier)
A =: All features
B =: All features but color moments
C =: All features but RGB histograms
D =: All features but LAB histogram
E =: All features but HSV histogram
F =: All features but texture features
G =: All features but variegation
Select k best
PCA
Select from model
All features with Yeo-Johnson normalization
Visually selected features (before normalization)
Visually selected features (after normalization)
A =: All features
H =: Best features based on the results [B - G]
I =: Sci-kit learn feature selection result
J =: All features with Yeo-Johnson normalization
K =: Visually selected features (before normalization)
L =: Visually selected features (after normalization)
Best Features

# Model Selection

# Challenge 1

The best model is selected based on the following evaluation metrics:

$$\kappa = \frac{2 \times (TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)}$$

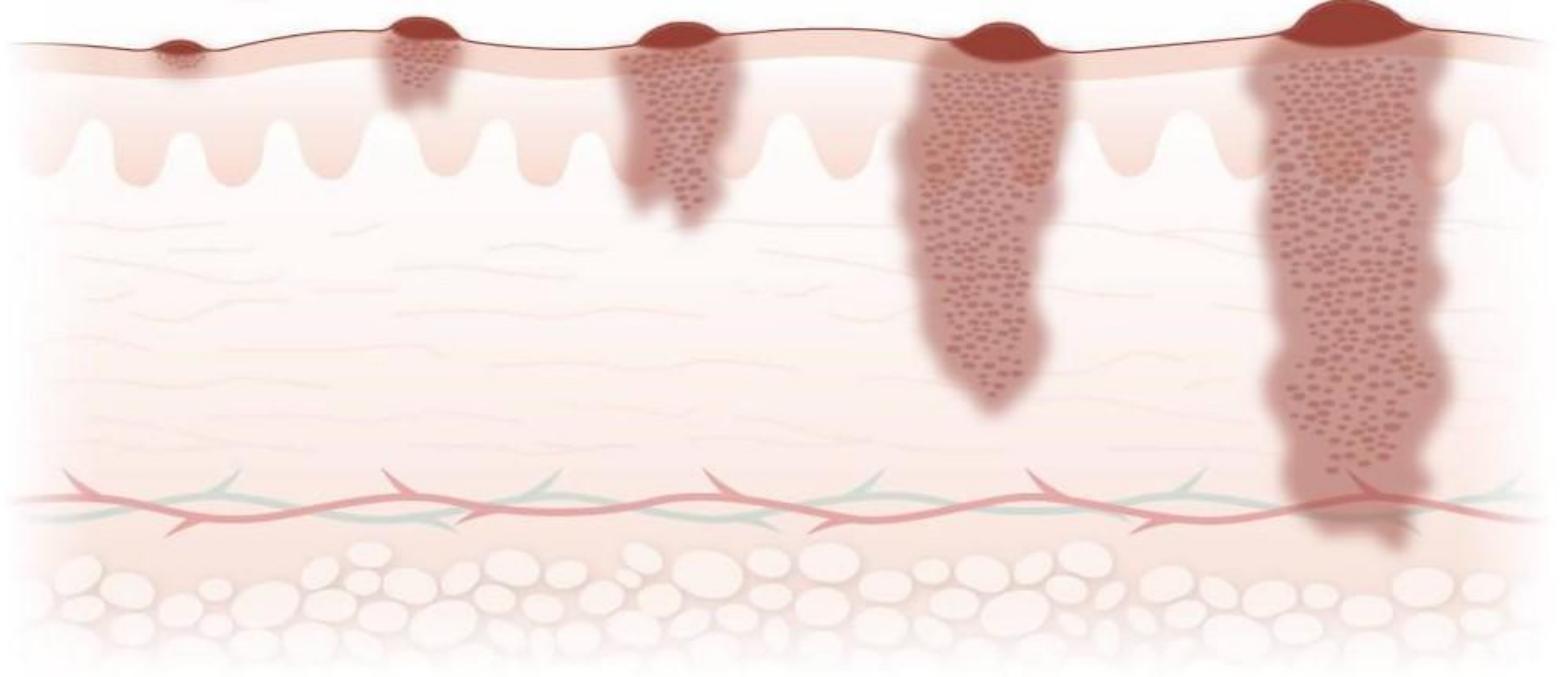
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

| | | True Labels | |
|---|---|---|---|
| | | Nevus | Other |
| Predicted Labels | Nevus | TP | FP |
| | Other | FN | TN |

# Challenge 1 (Cont'd)

The following models were compared:

- Nearest Neighbor (Baseline)
- K-Nearest Neighbors
- Support Vector Machines
- Gradient Boosting (XGBoost)
- Extra Trees
- Random Forest
- Stacking ensemble (SVM + XGboost + Extra trees with Logistic Regression as meta classifier)
- Averaging ensemble (SVM + XGboost + Extra trees)
- Majority voting ensemble (SVM + XGboost + Extra trees)
- Ensembling the pre-trained models with averaging
- Ensembling the pre-trained models with majority voting

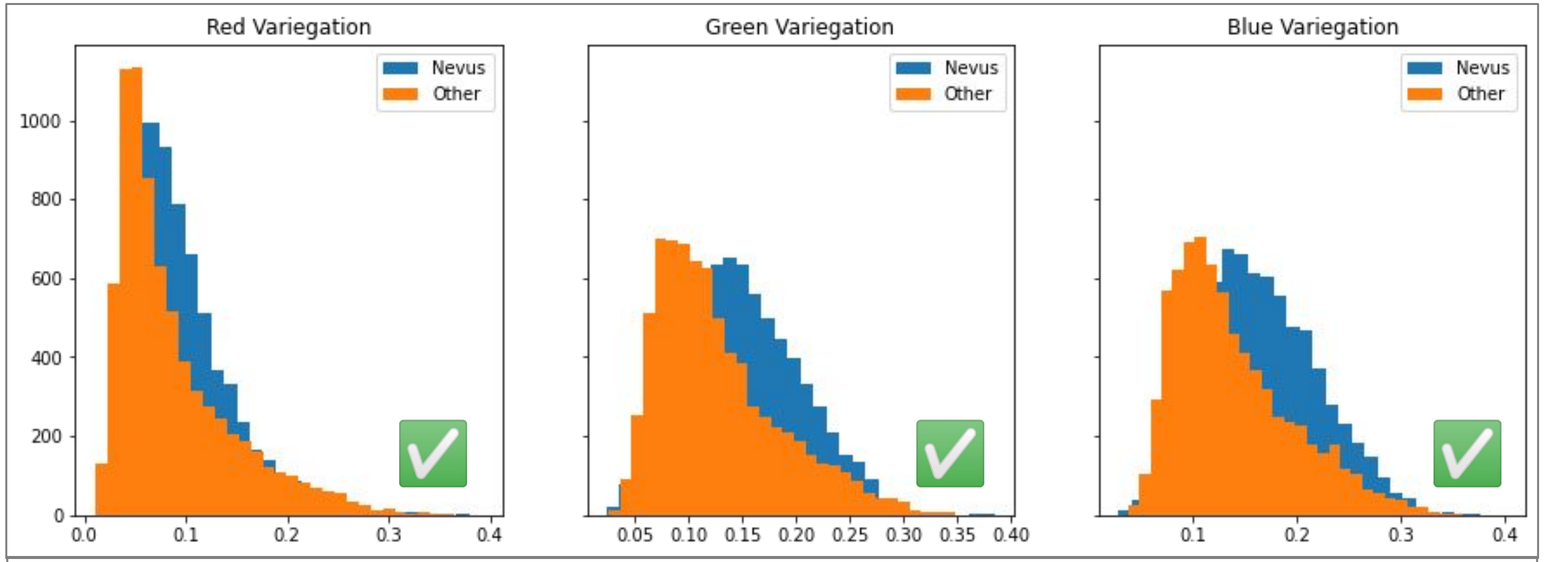


# Experimental Results

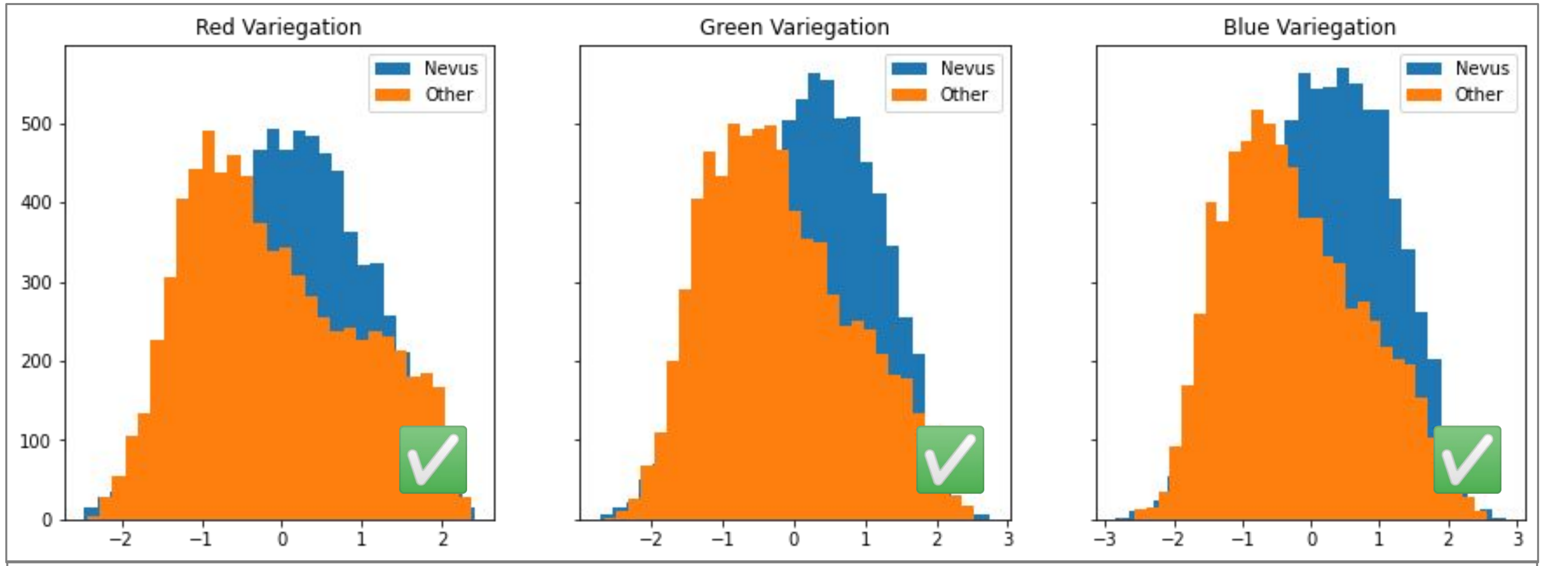
# Feature Visualization

With MatPlotLib

# Variegation

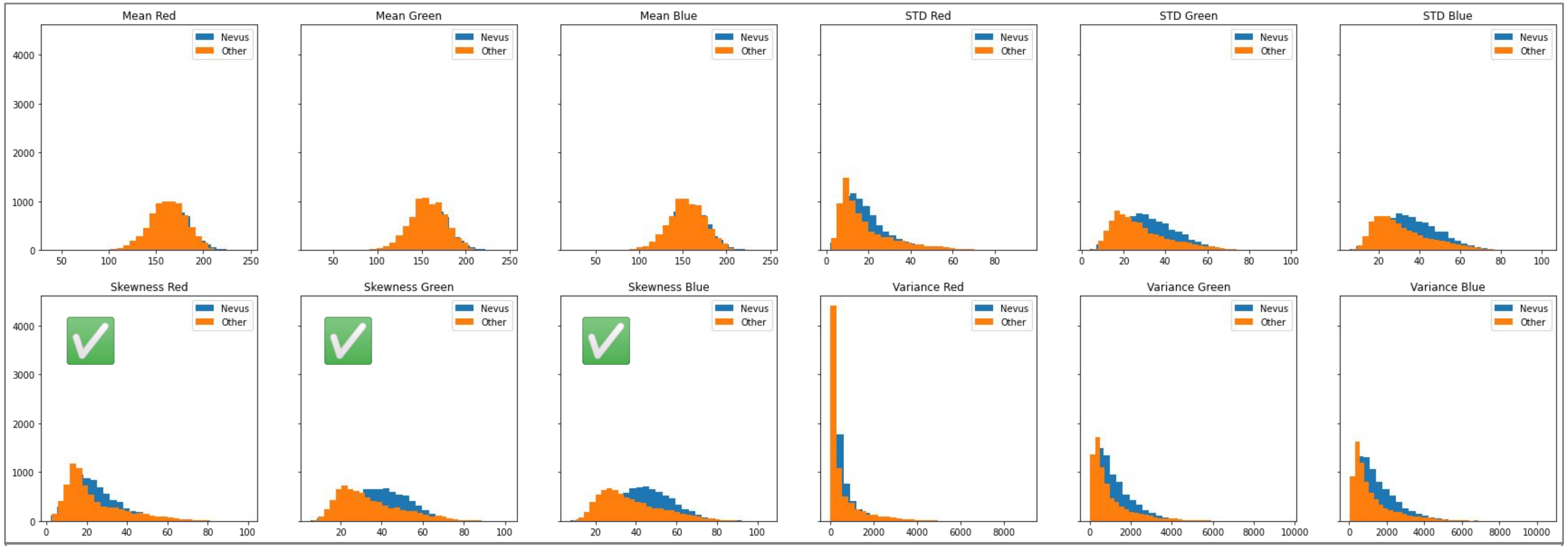



Before Normalization

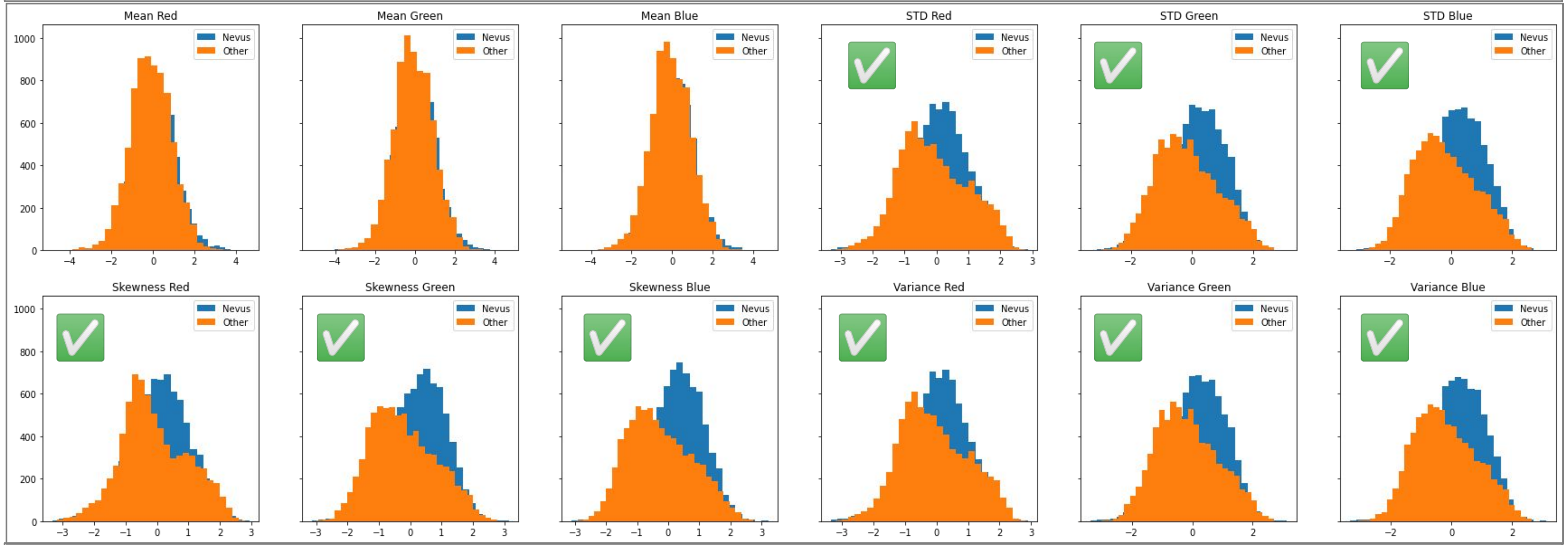



After Normalization

# Color Moments

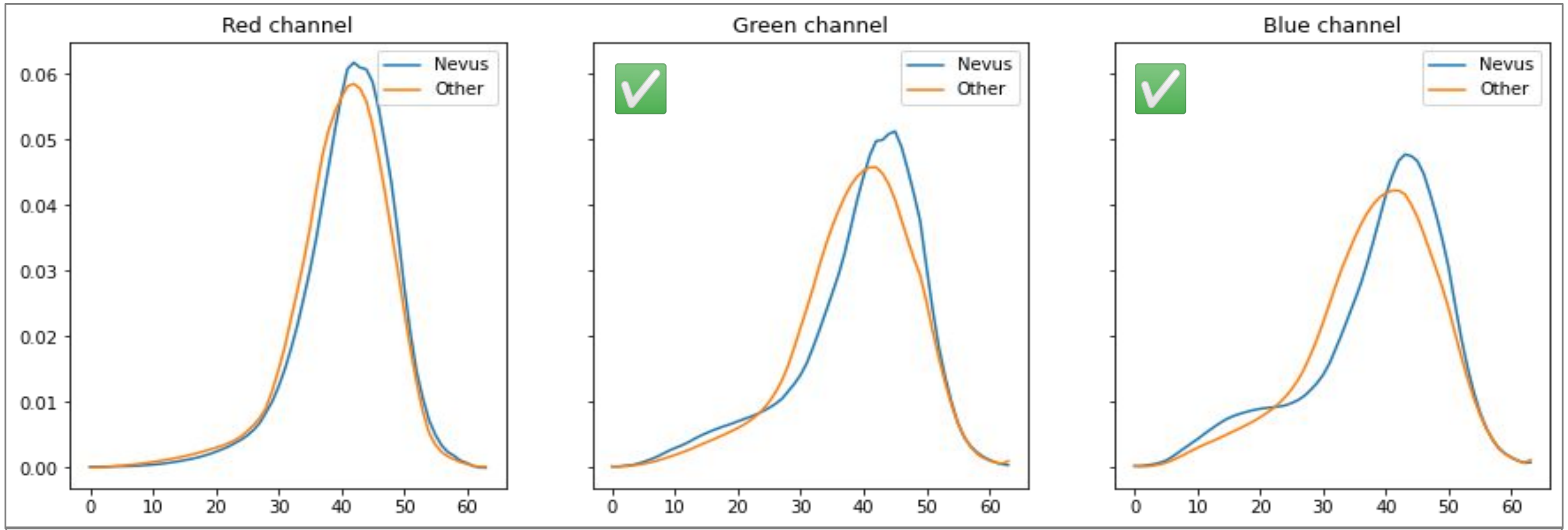



Before Normalization

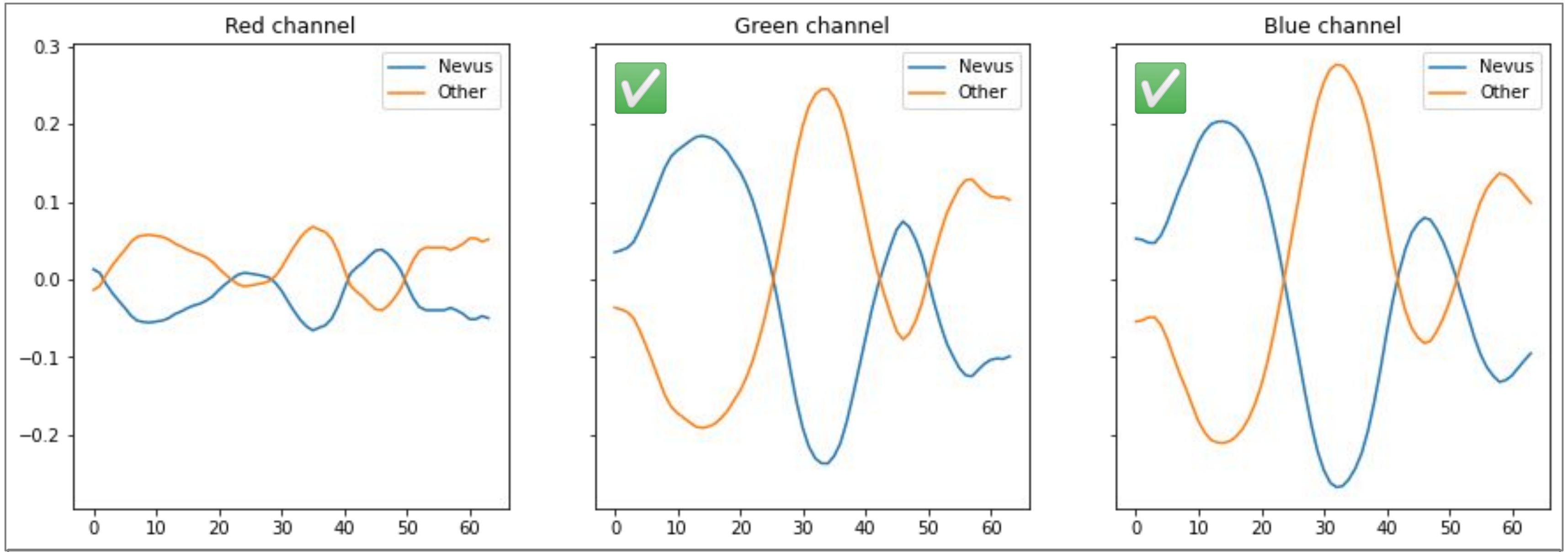



After Normalization

# Mean RGB Histograms For Each Class

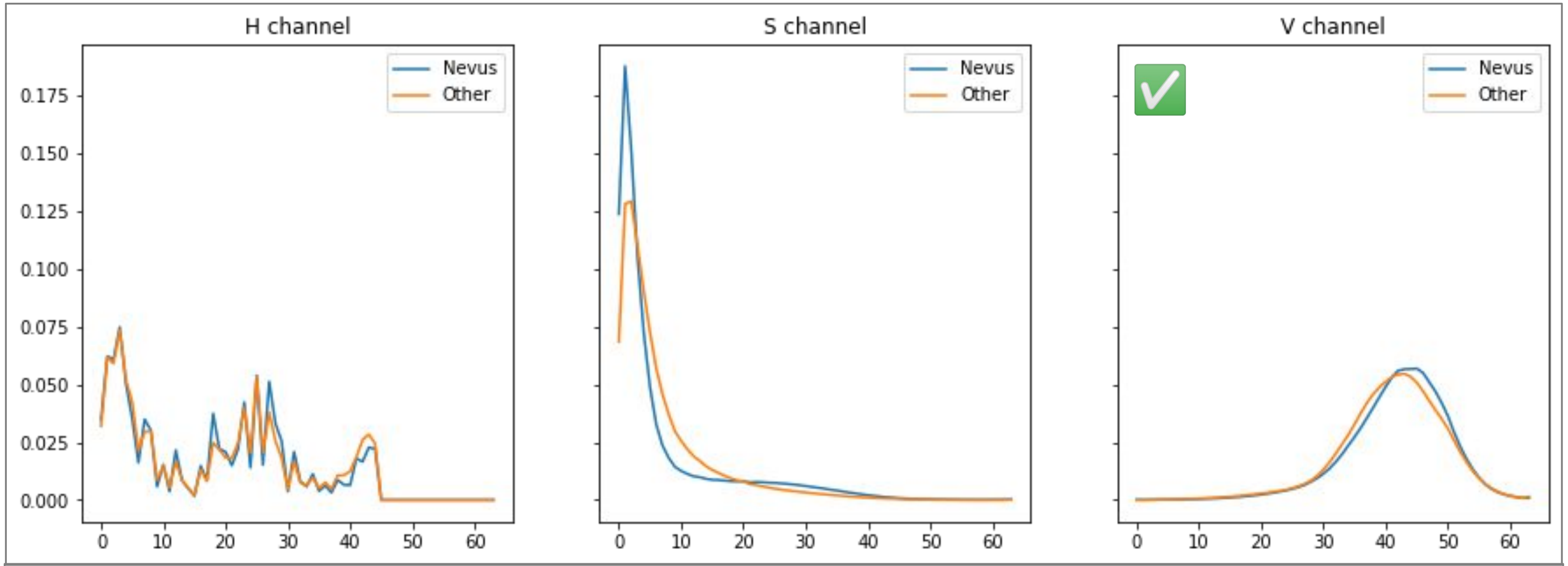



Before Normalization

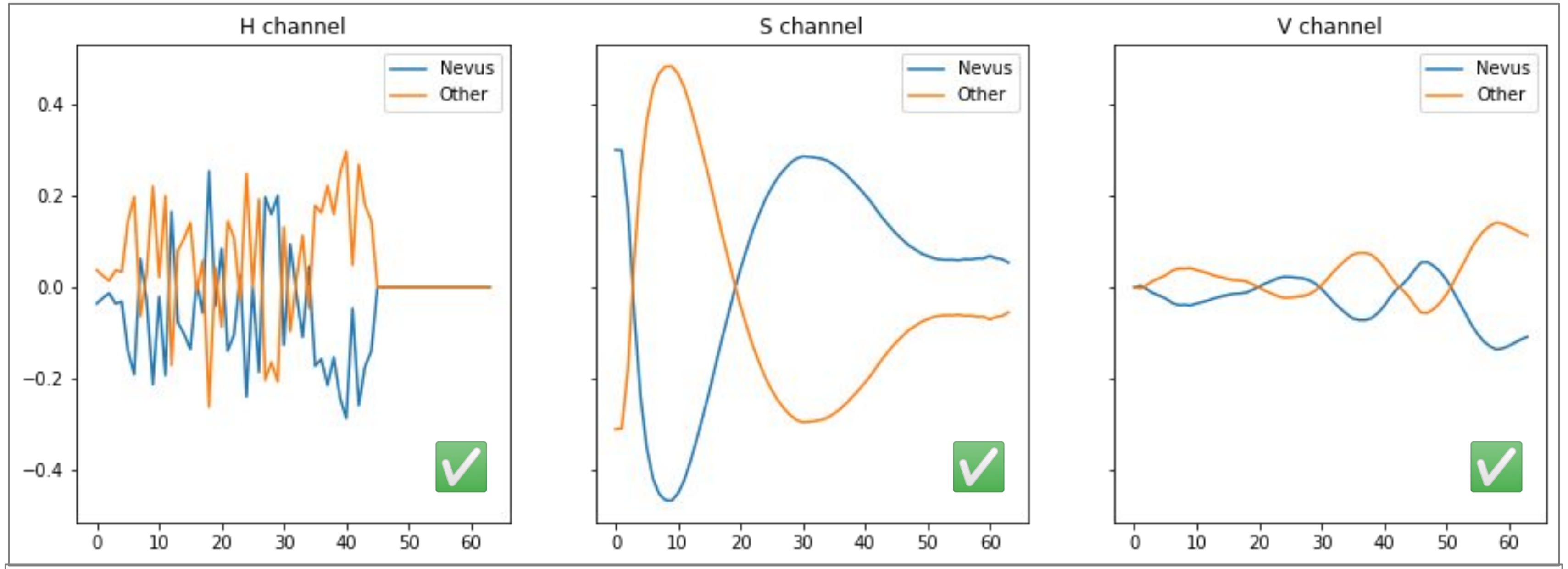



After Normalization

# Mean HSV Histograms For Each Class

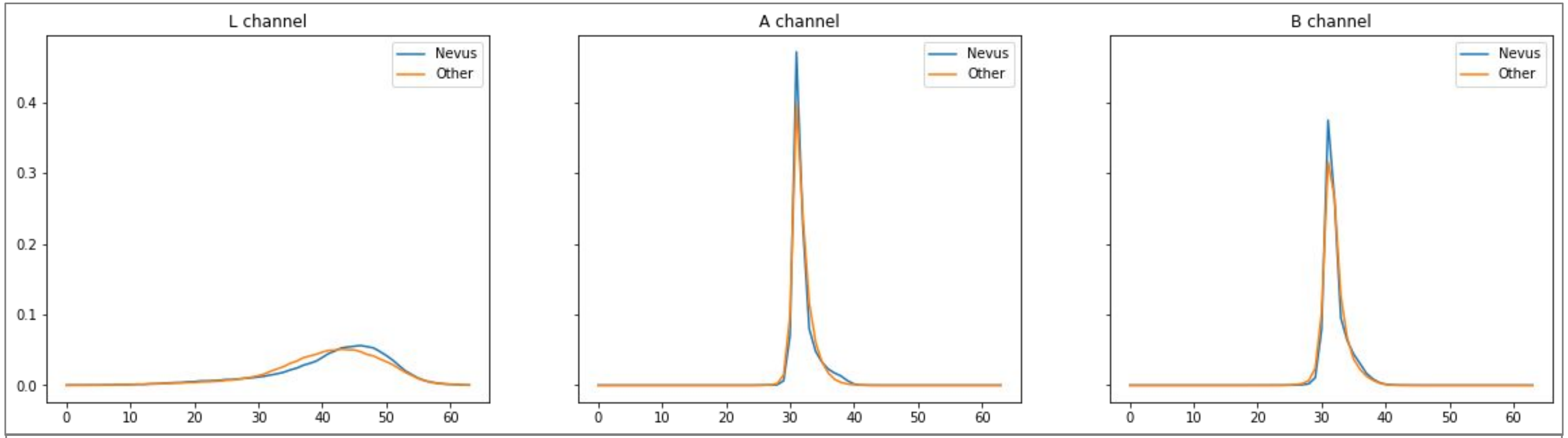



Before Normalization

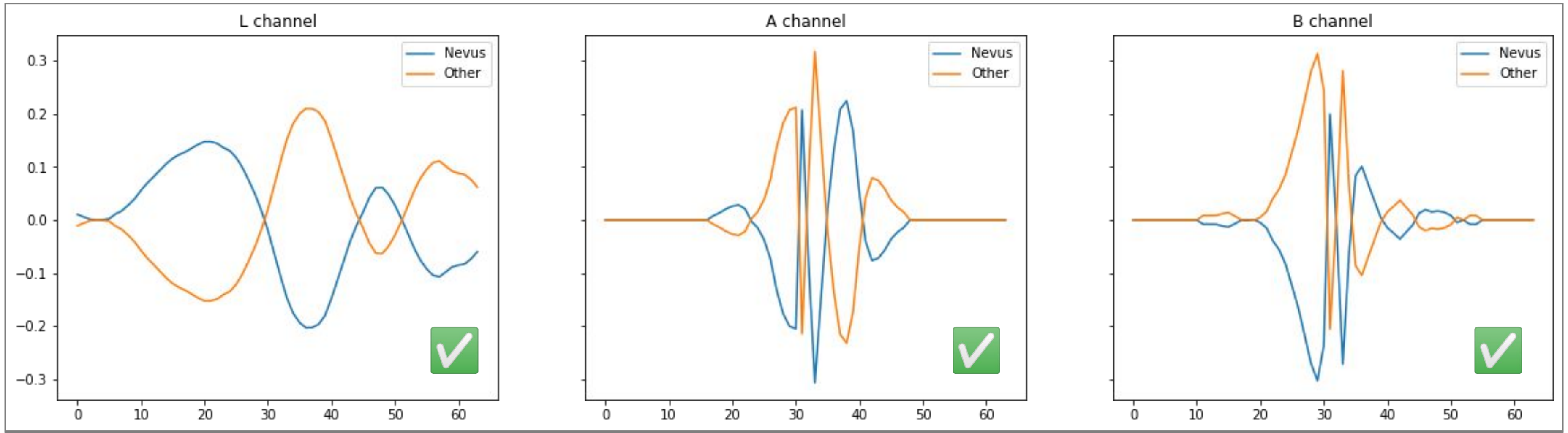



After Normalization

# Mean LAB Histograms For Each Class

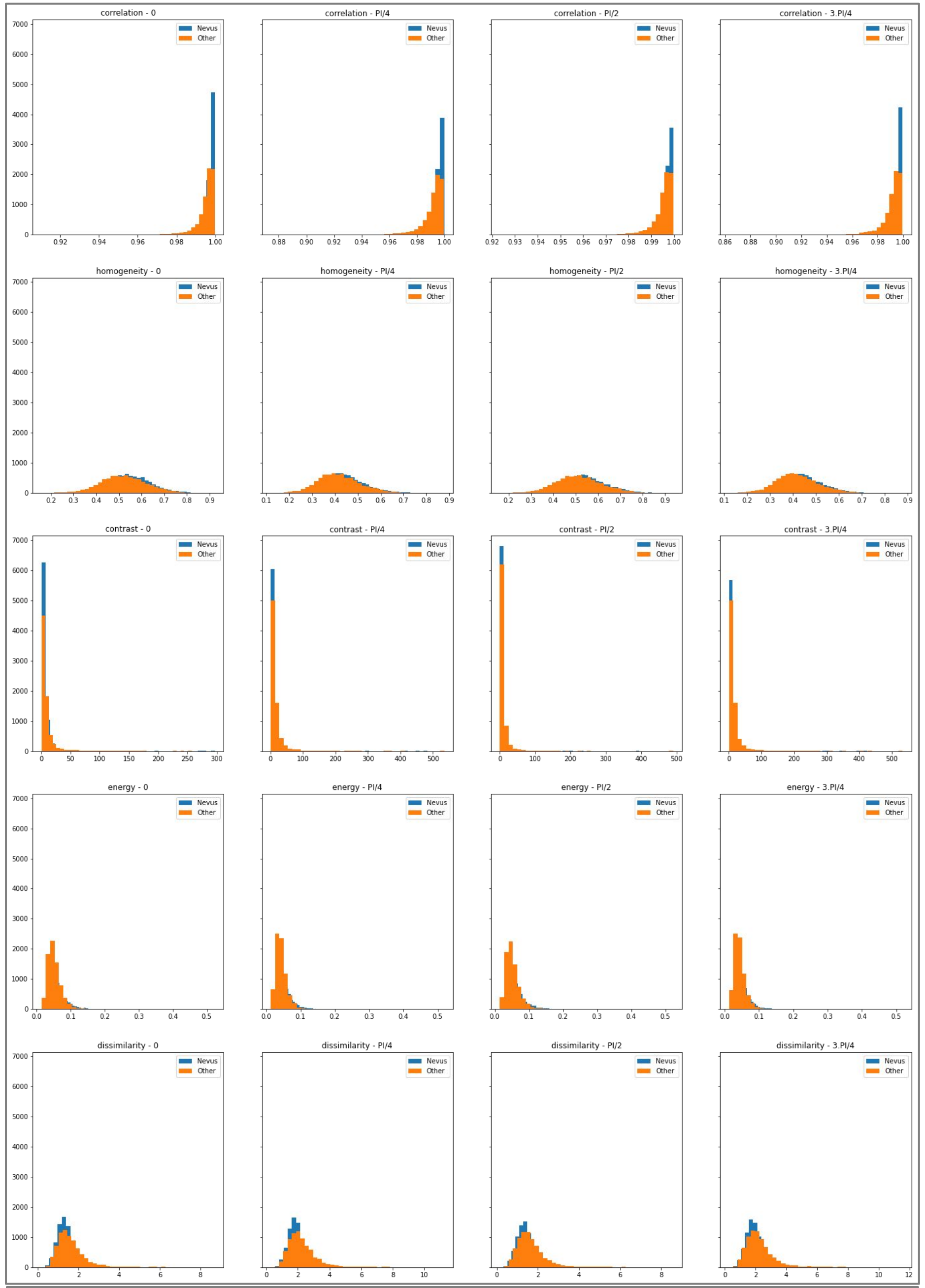



Before Normalization

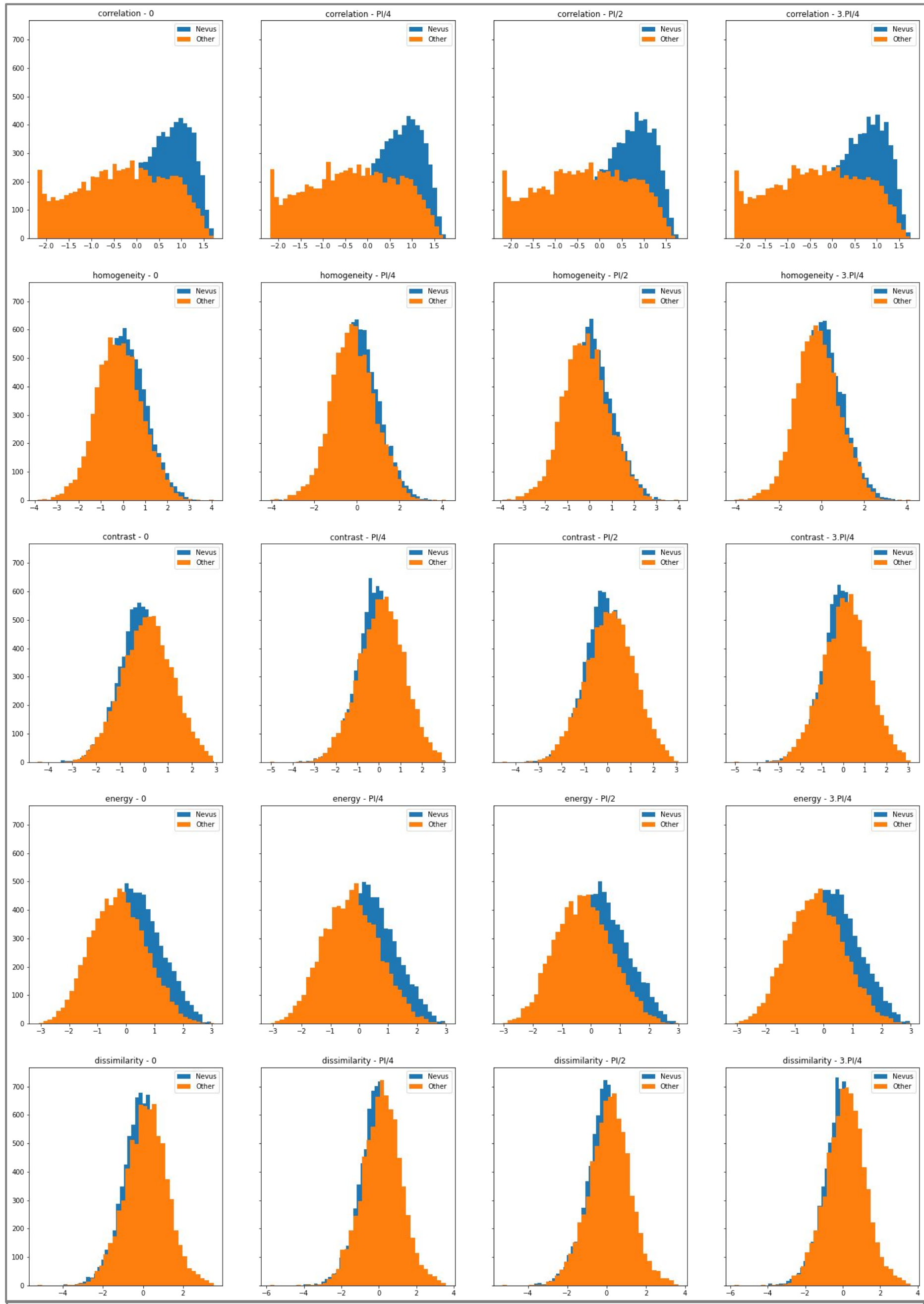



After Normalization

# GLCM Features

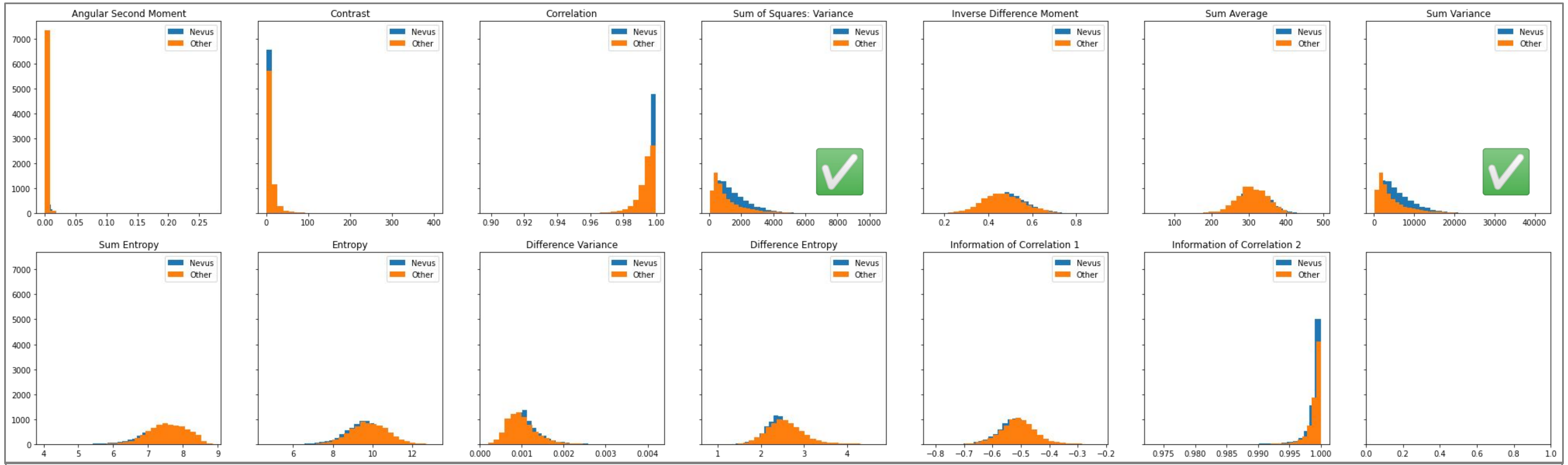



Before Normalization

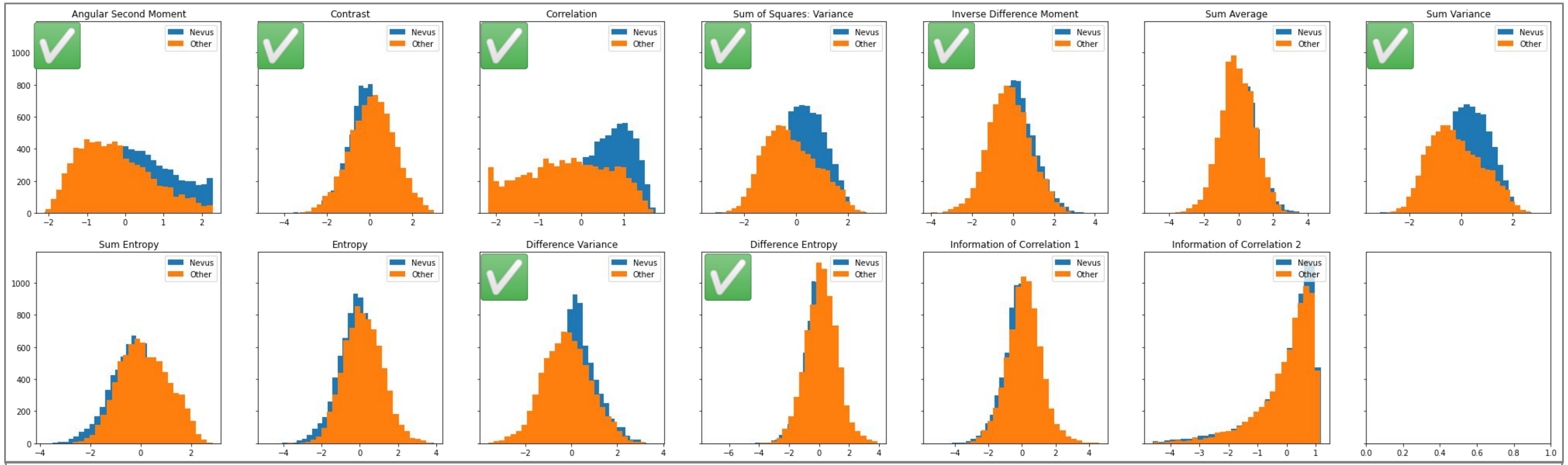



After Normalization

# Haralick Features

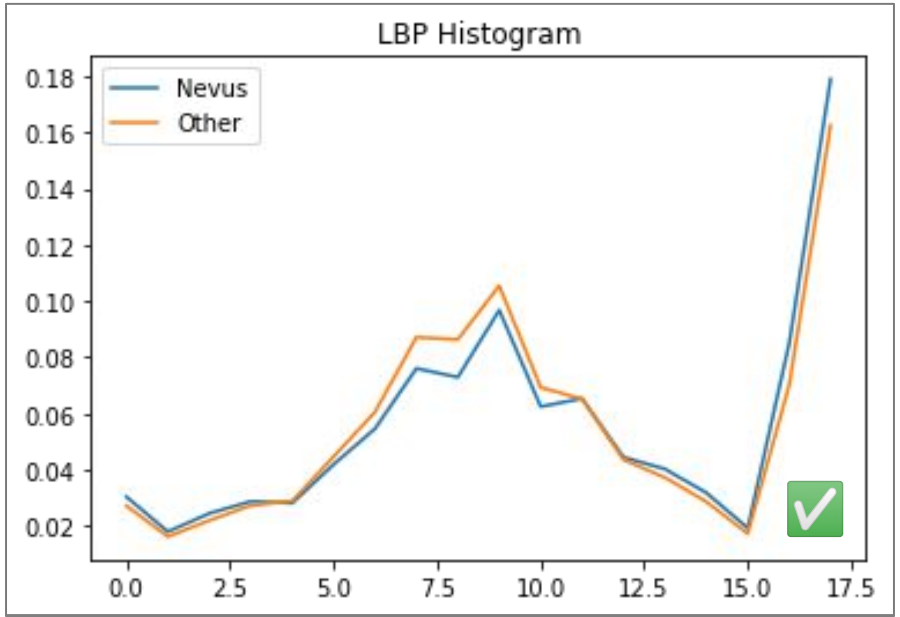



Before Normalization

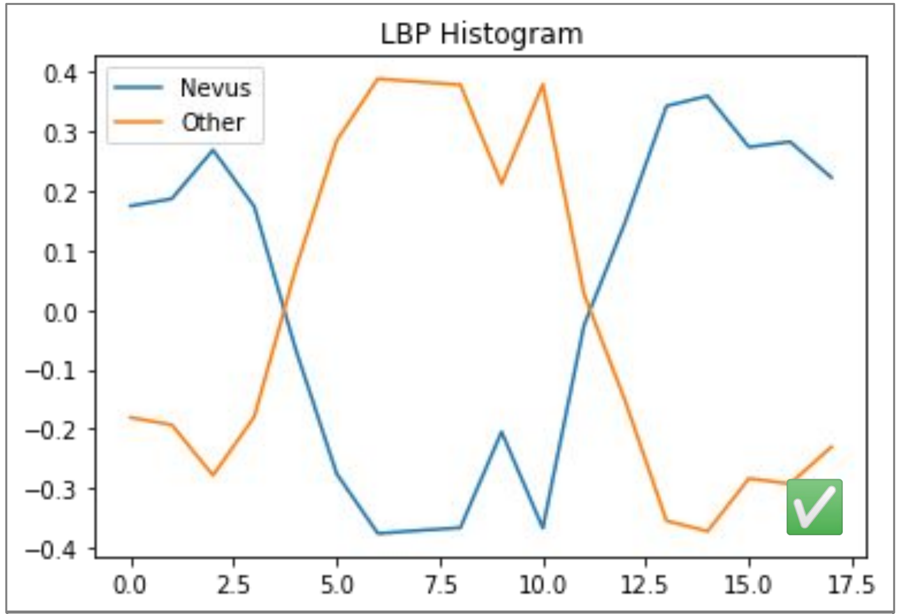



After Normalization

# LBP Histograms




Before Normalization

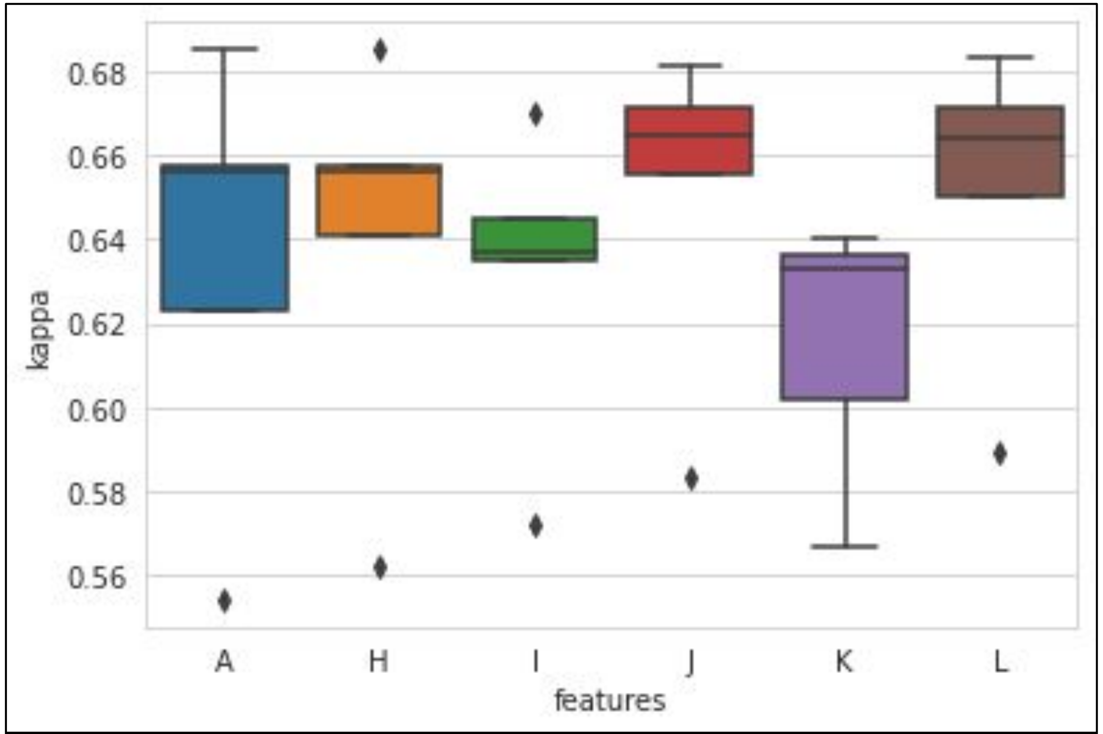



After Normalization

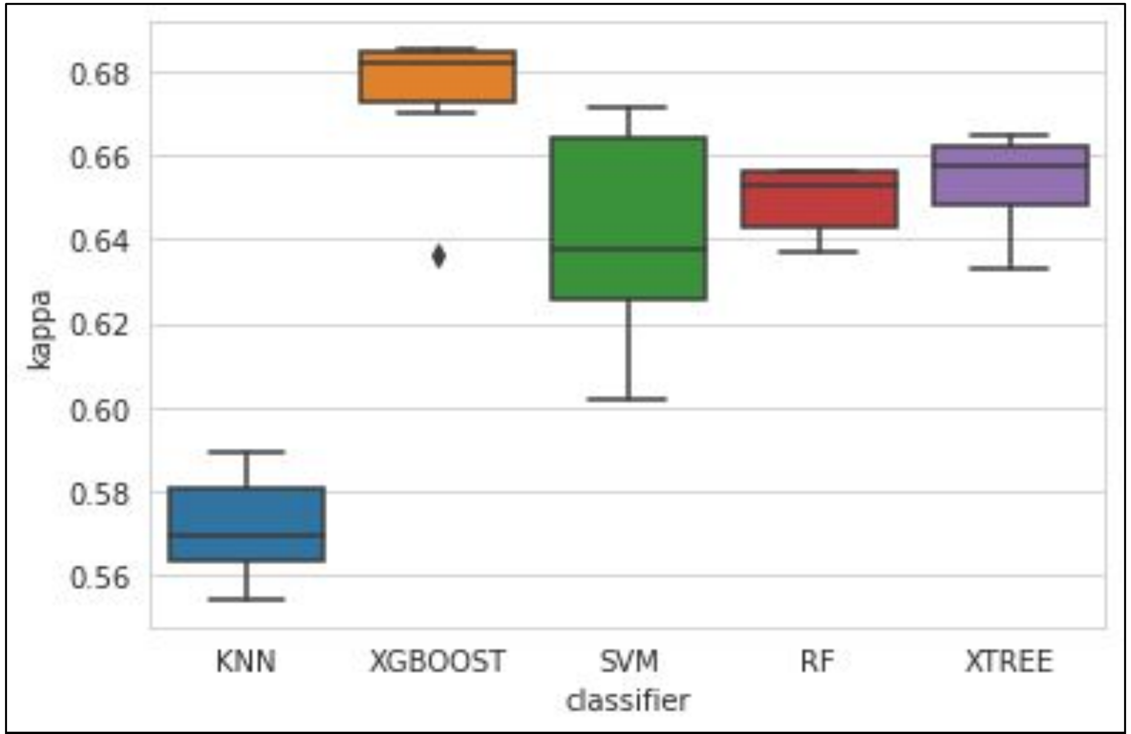
# Training and Evaluation

Feature selection & model training results

# Global Results Evaluation




Boxplot of the kappa metric for each feature vector across all models



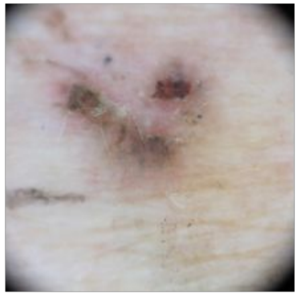

Boxplot of the kappa metric for each classifier across all feature vectors

# Best Results Per Classifier

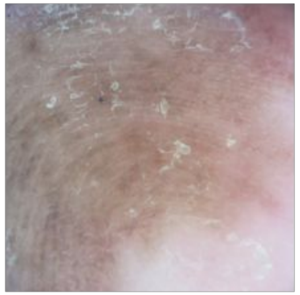
| Classifier | Parameters | Features | Time | Accuracy | Kappa |
|---|---|---|---|---|---|
| K-NN | k=20 | L | 0035.718 | 0.794783 | 0.589447 |
| XGboost | n_estimators = 2000, depth = 9, lr = 0.1 | A | 2214.809 | 0.842729 | 0.685325 |
| SVM | C = 8 | L | 0318.782 | 0.835879 | 0.671699 |
| RF | n_estims = 500, depth = 20, criterion = gini | A | 0135.674 | 0.828240 | 0.656174 |
| Extra Trees | n_estims = 2000, depth = 9, criterion = gini | J | 0080.913 | 0.832718 | 0.665179 |
| Stacking | XGboost + SVM + Extra Trees | L | 10339.71 | 0.840358 | 0.680631 |
| Averaging (sklearn) | XGboost + SVM + Extra Trees | L | 2687.811 | 0.842992 | 0.685815 |
| Majority Voting (Sklearn) | XGboost + SVM + Extra Trees | L | 2564.247 | 0.843256 | 0.686305 |
| **Averaging (manual)** | **XGboost + SVM** | **L** | **–** | **0.844573** | **0.689023** |
| Majority Voting (manual) | XGboost + SVM + Extra Trees | L | – | 0.844046 | 0.687884 |

# Best Classifier Results

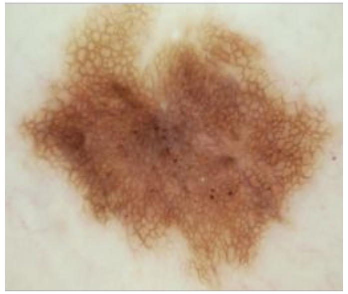
bcc02036: Others



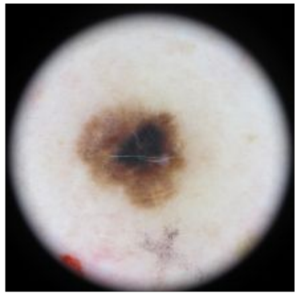
mel03075: Others



nev07849: Nevus

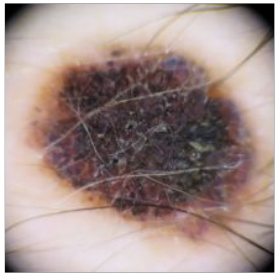


nev08188: Nevus

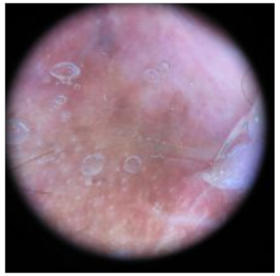


Examples of Correctly Classified Images

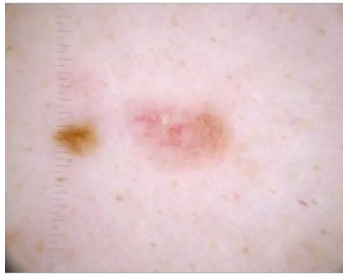
nev08216: Others



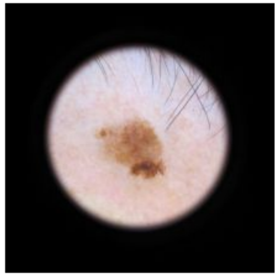
nev09099: Others



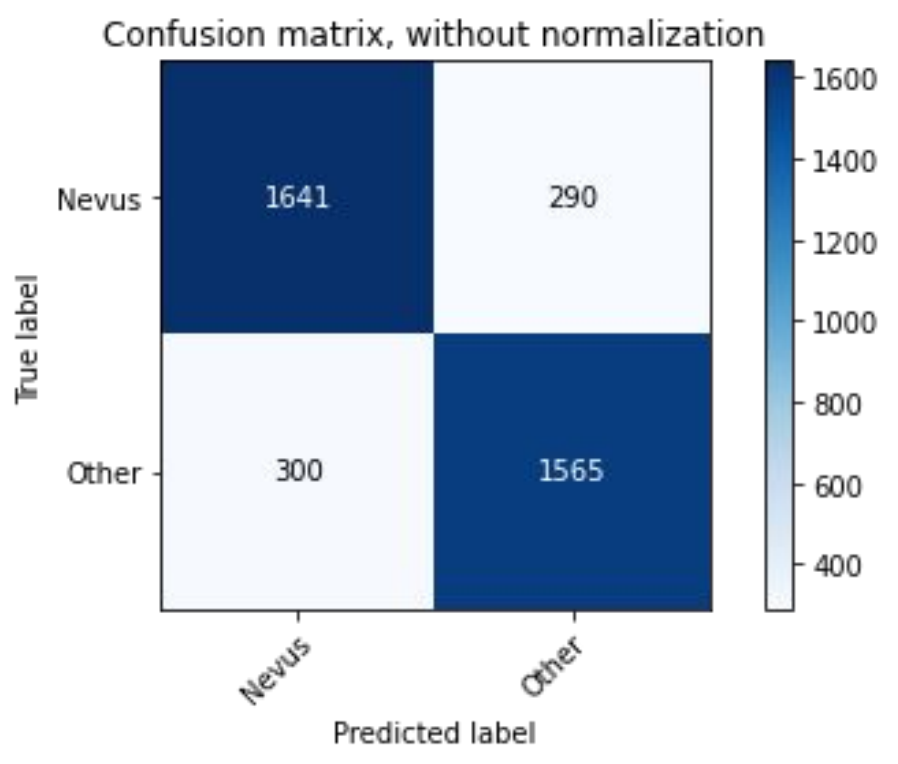
bcc02119: Nevus



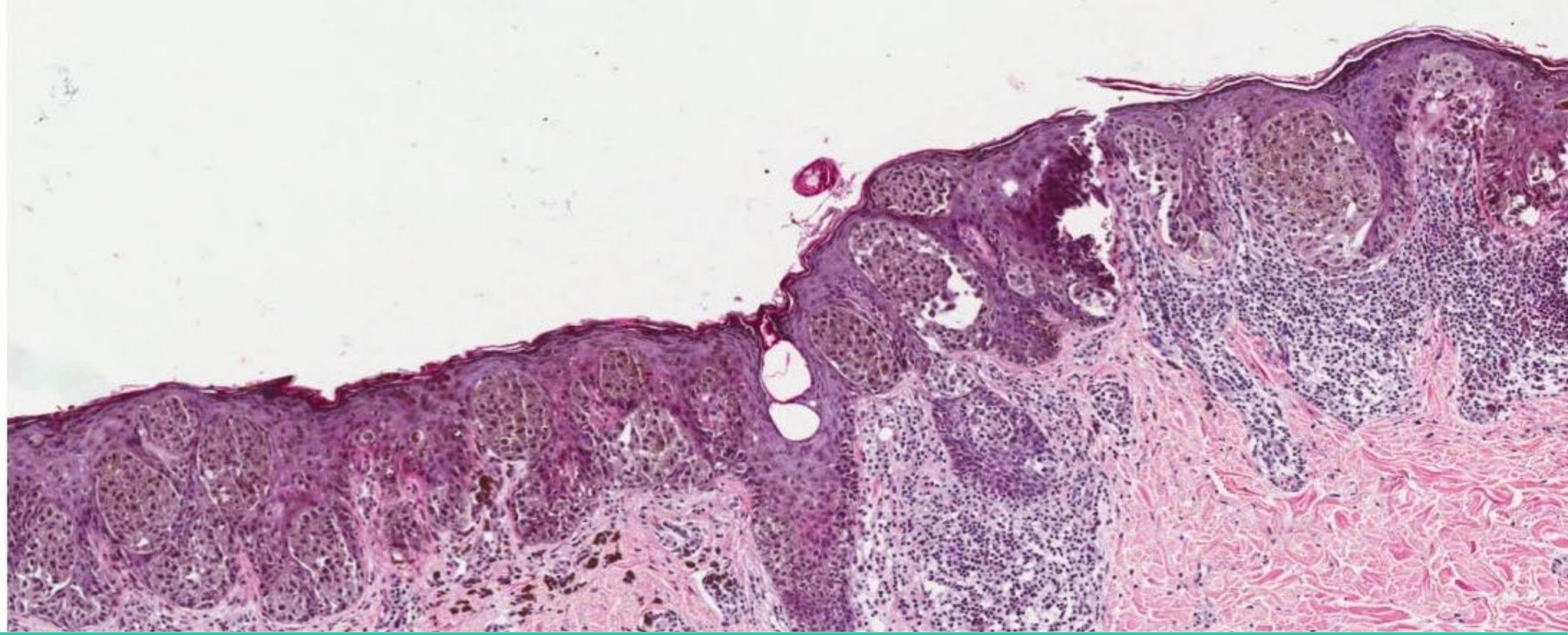
ack00606: Nevus



Examples of Wrongly Classified Images



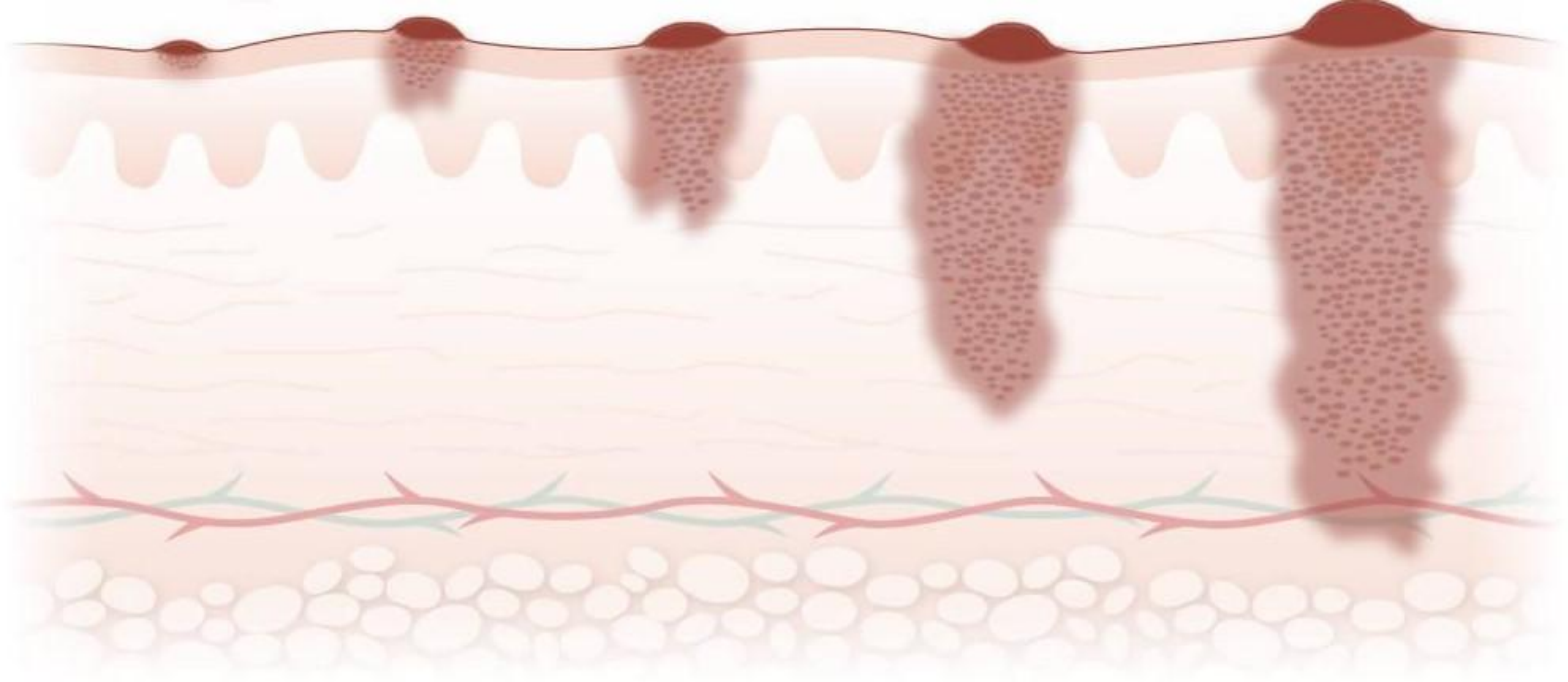

Confusion matrix on the validation set



# Challenge 2: Melanoma VS BCC VS SCC

# Notes

- For this challenge, we follow the same proposed pipeline as in Challenge 1.
- The same preprocessing steps were applied.
- The same features were extracted.

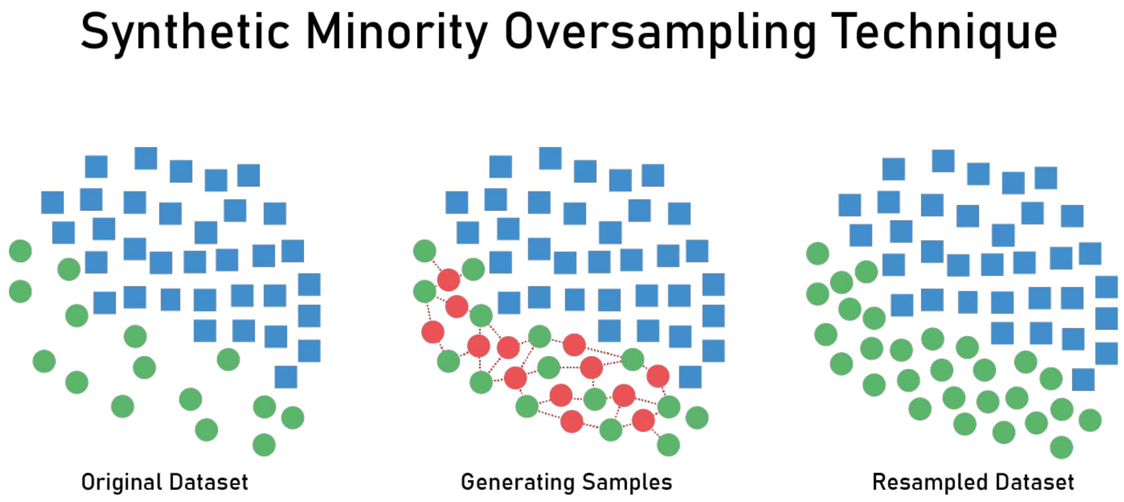


# Sampling

# Class imbalance with SMOTE and Undersampling

For Challenge 2, class imbalance is tackled by sampling: Oversampling the minority class (~1700 samples) with SMOTE, then randomly undersampling the other classes to that same amount.

1. SMOTE : an oversampling technique used on minority class. It works by selecting samples from the minority class and finding the k-nearest neighbors of each of them. Then, synthetic examples are created at a randomly selected point between two of those neighbors.
2. Random Undersampling : Randomly selecting samples from the majority classes without replacement, while keeping the minority class unchanged.



# Feature Selection

## Feature Selection Pipeline (For each classifier)

- A =: All features
- B =: All features but color moments
- C =: All features but RGB histograms
- D =: All features but LAB histogram
- E =: All features but HSV histogram
- F =: All features but texture features
- G =: All features but variegation

- Select k best
- PCA
- Select from model

- All features with Standard Scaler preprocessing
- Visually selected features (after normalization)

- A =: All features
- H =: Best features based on the results [B - G]
- I =: Sci-kit learn feature selection result
- J =: All features with Standard Scaler preprocessing
- L =: Visually selected features (after normalization)

**Best Features**

# Model Selection

# Challenge 2

The best model is selected based on the following evaluation metrics:

$$\text{Kappa} = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

$$\texttt{balanced-accuracy} = \frac{1}{3}\left(\frac{TP}{TP+FN}\right)$$

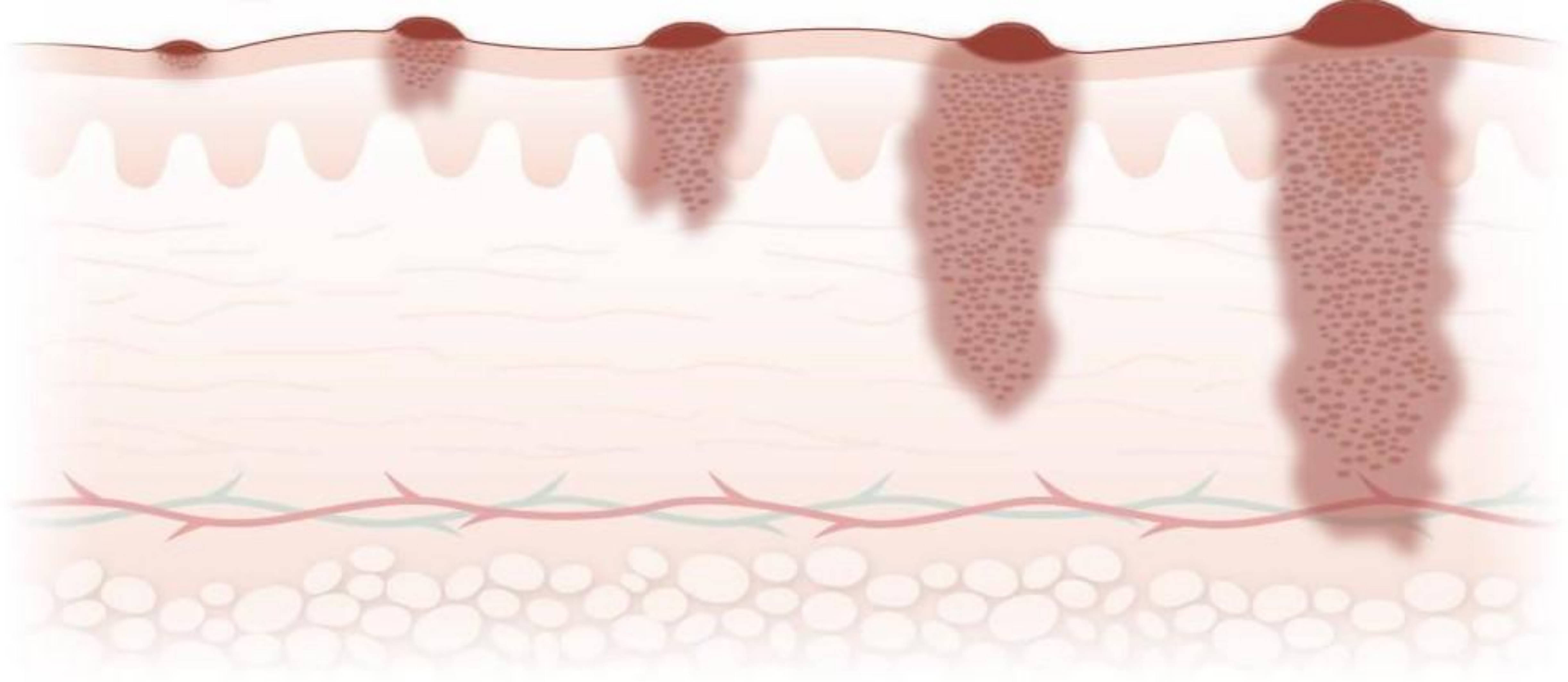
Predicted Labels

| True Labels | Mel | Bcc | Scc |
|---|---|---|---|
| Scc | TN | FP | TN |
| Bcc | FN | TP | FN |
| Mel | TN | FP | TN |

# Challenge 2 (Cont’d)

The following models were compared:

- Nearest Neighbor (Baseline)
- K-Nearest Neighbors
- Support Vector Machines
- Gradient Boosting (XGBoost)
- Extra Trees
- Random Forest
- Stacking ensemble
- Majority voting ensemble (soft)
- Ensembling the pre-trained models with averaging
- Hierarchical architecture with previous classifiers

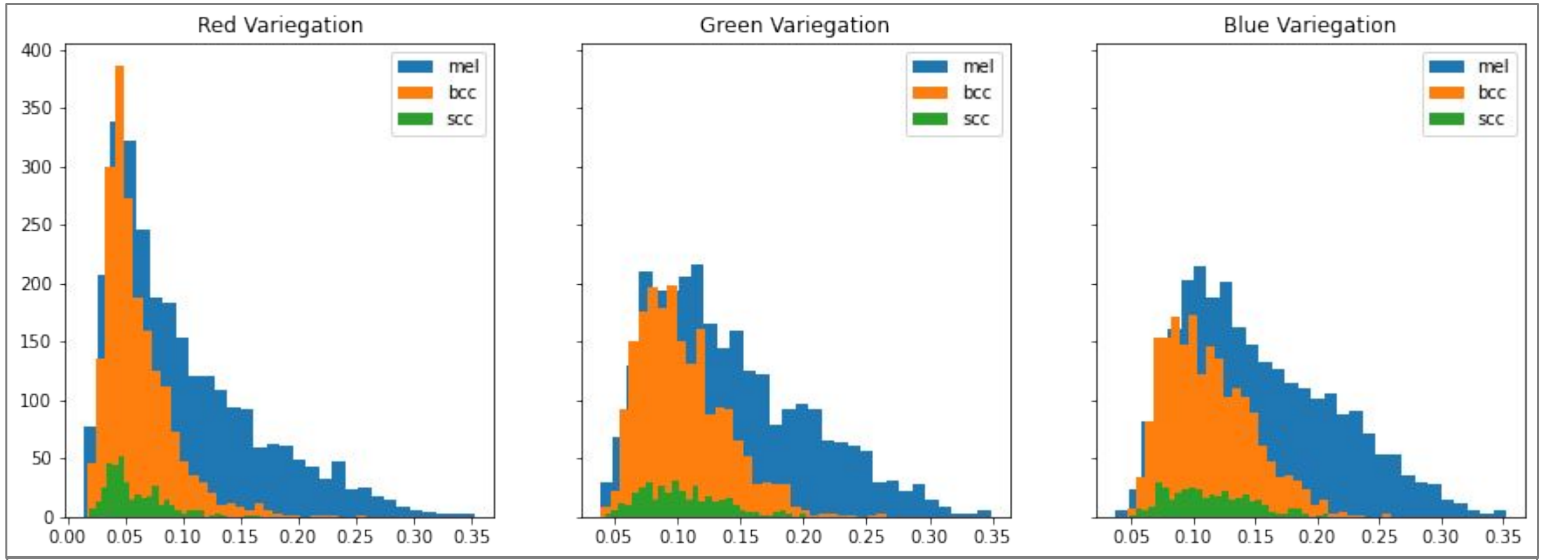


# Experimental Results

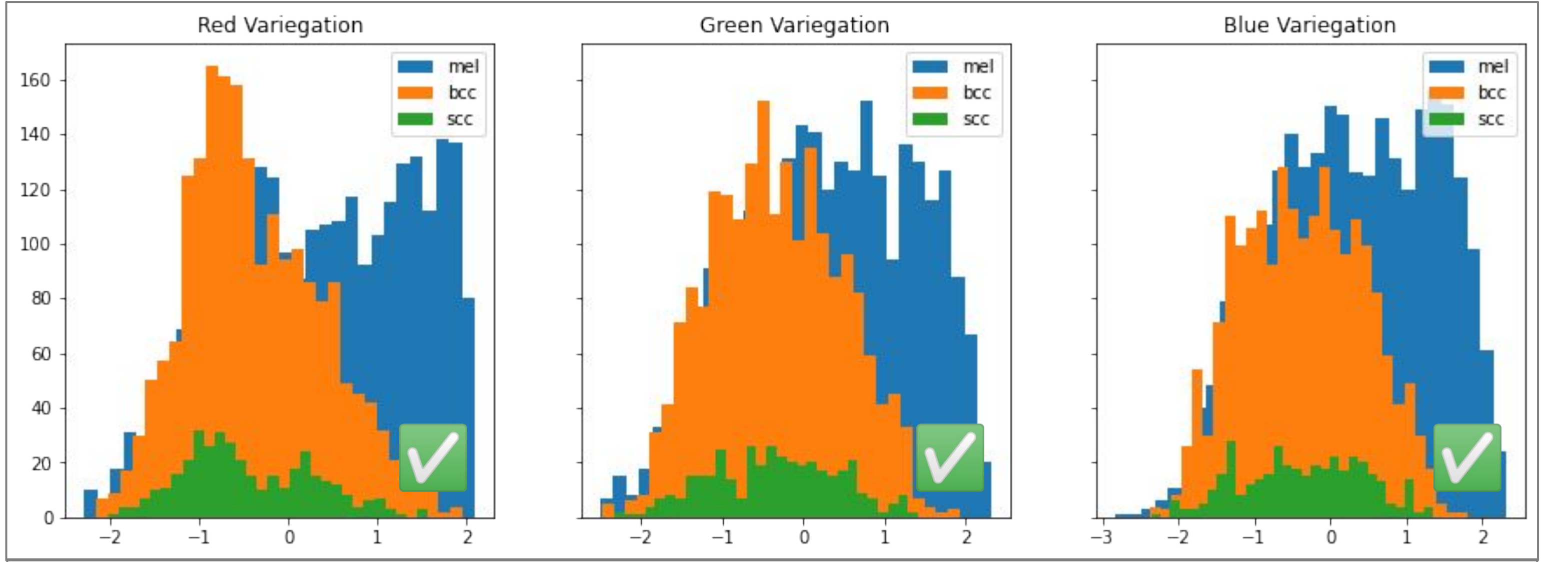
# Feature Visualization

With MatPlotLib

# Variegation

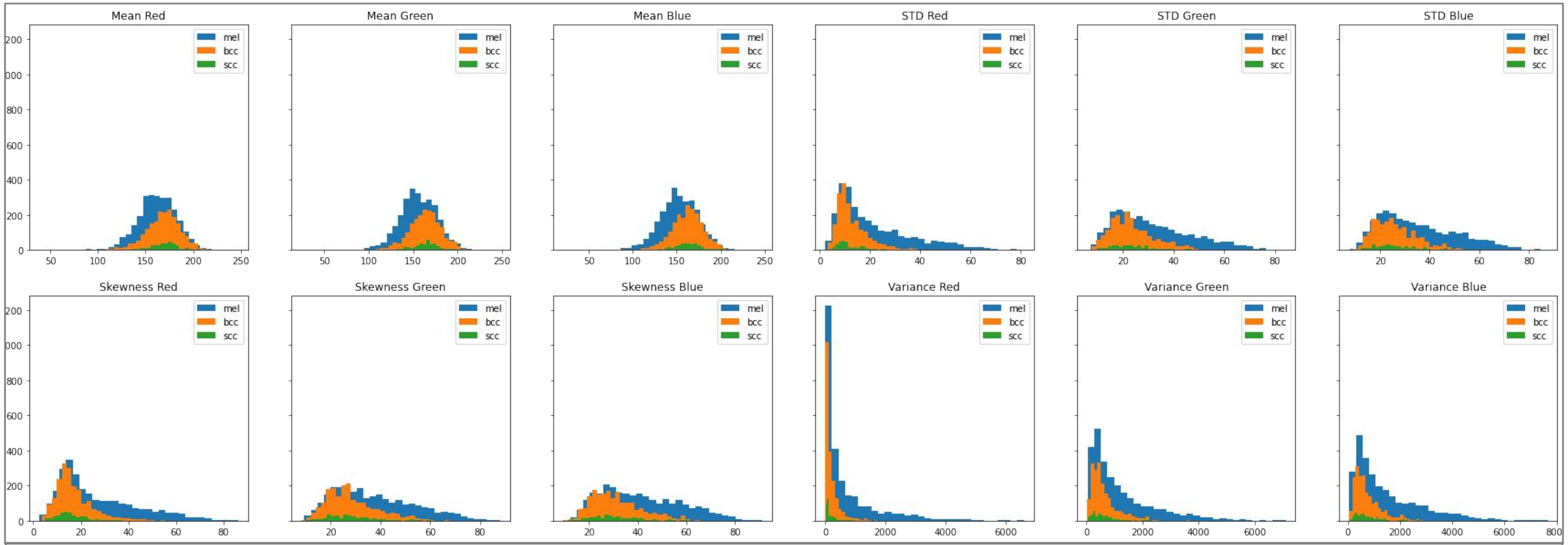



Before Normalization

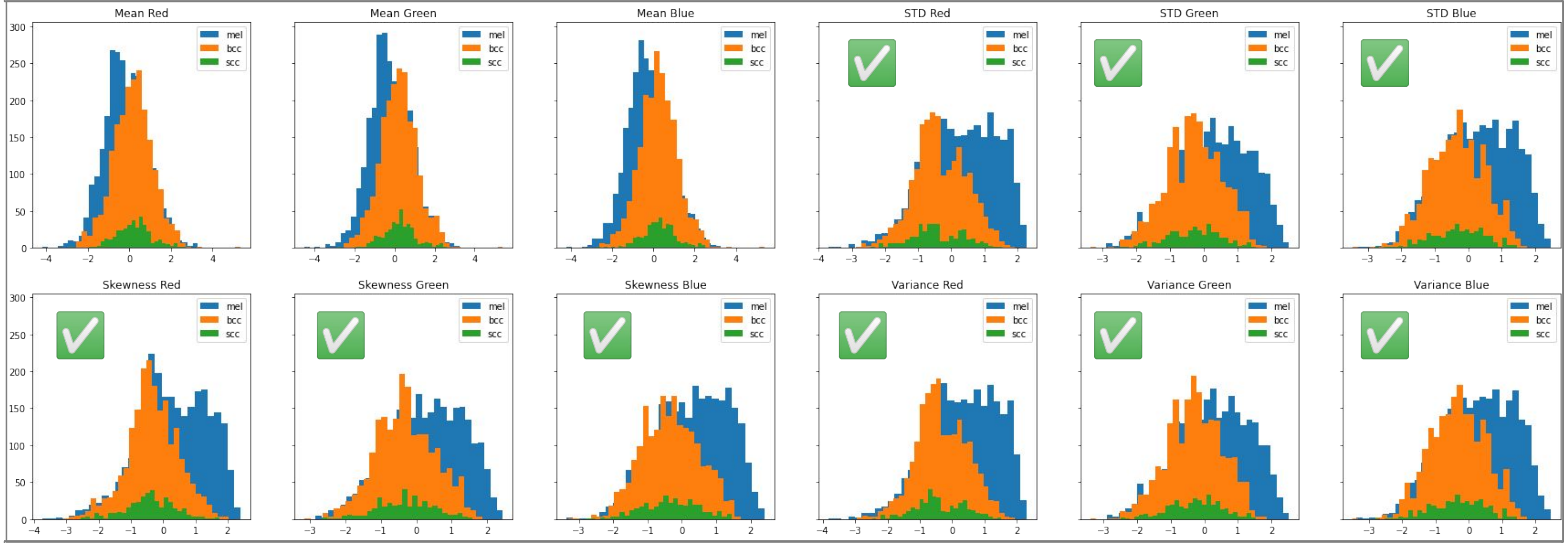



After Normalization

# Color Moments

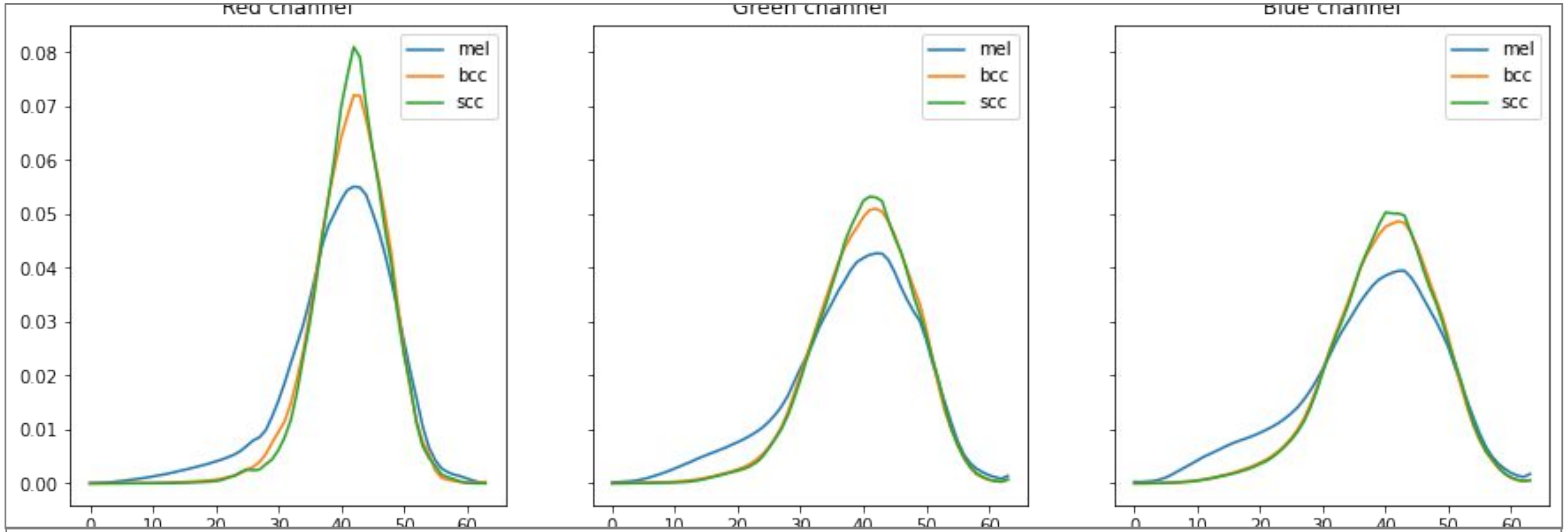



Before Normalization

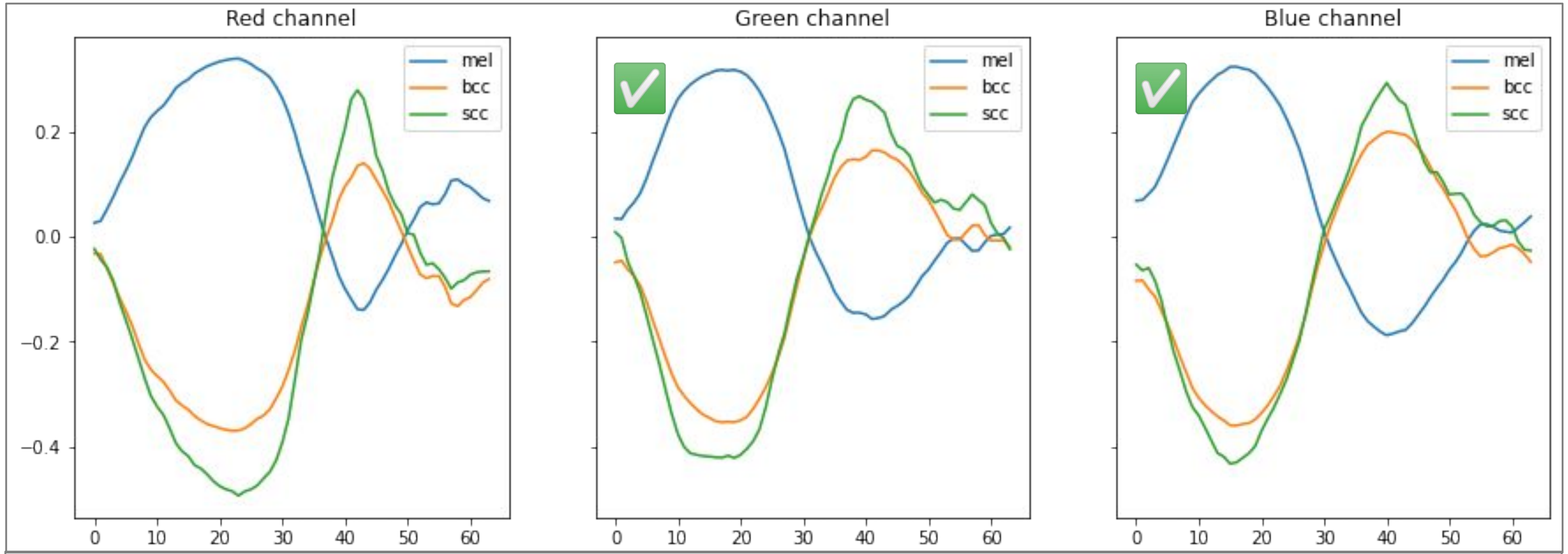



After Normalization

# Mean RGB Histograms For Each Class

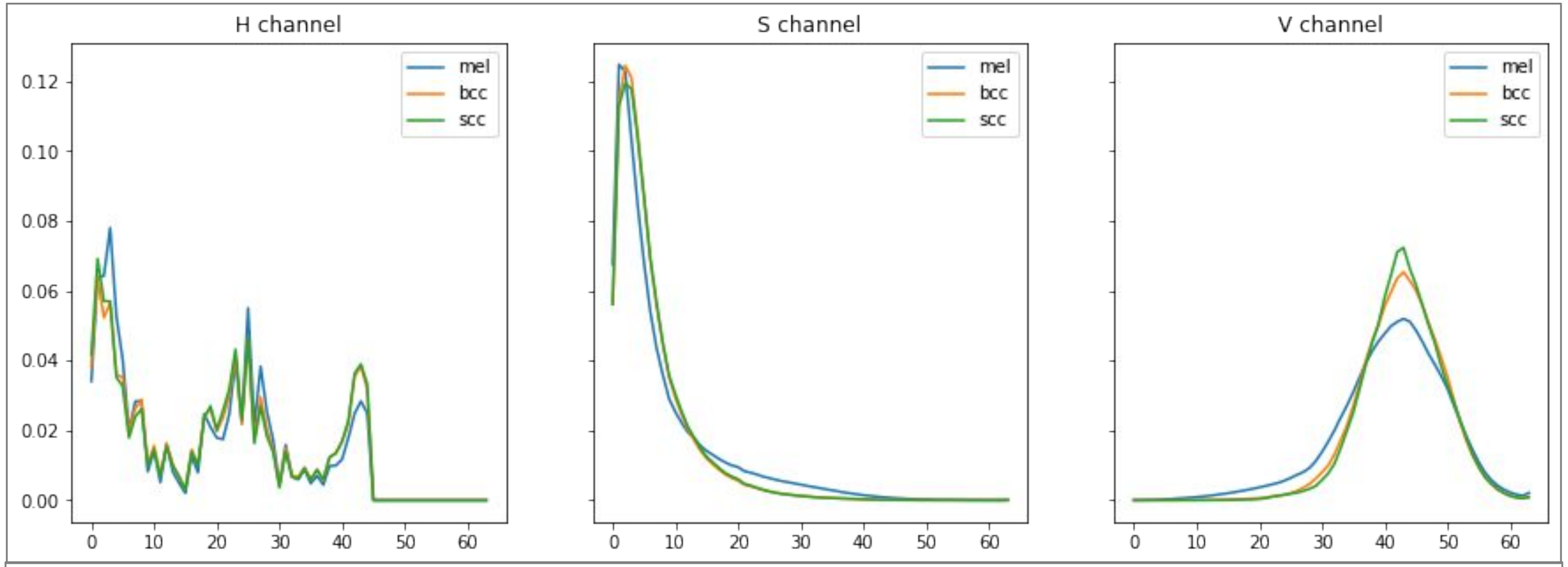



Before Normalization

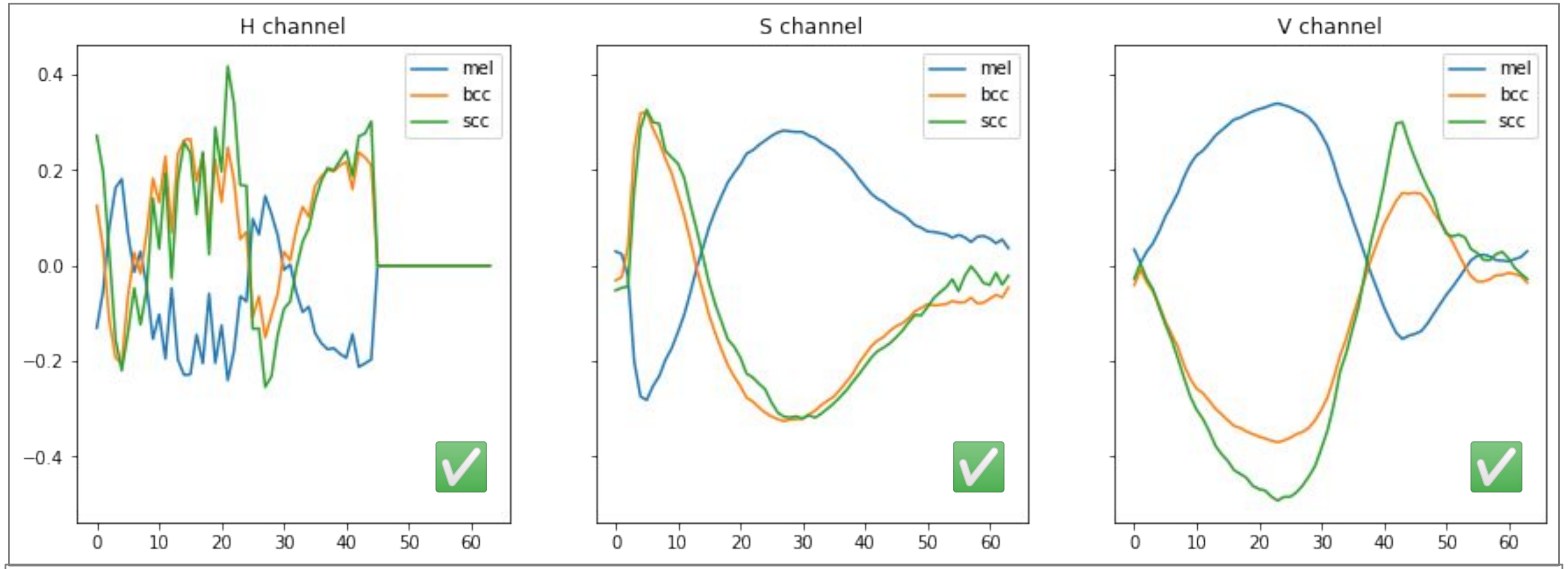



After Normalization

# Mean HSV Histograms For Each Class

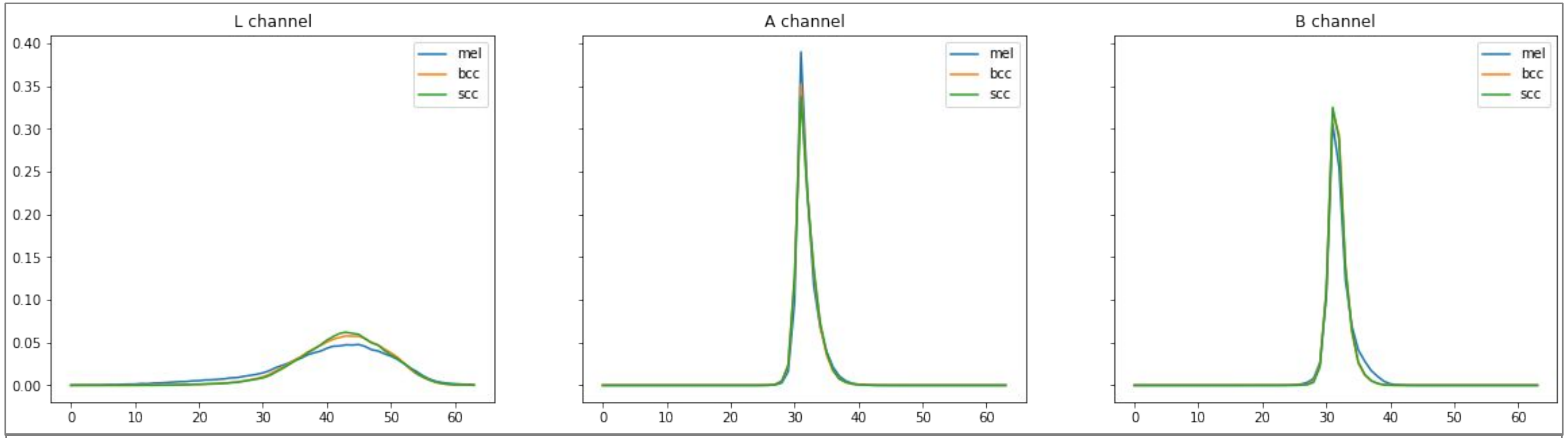



Before Normalization

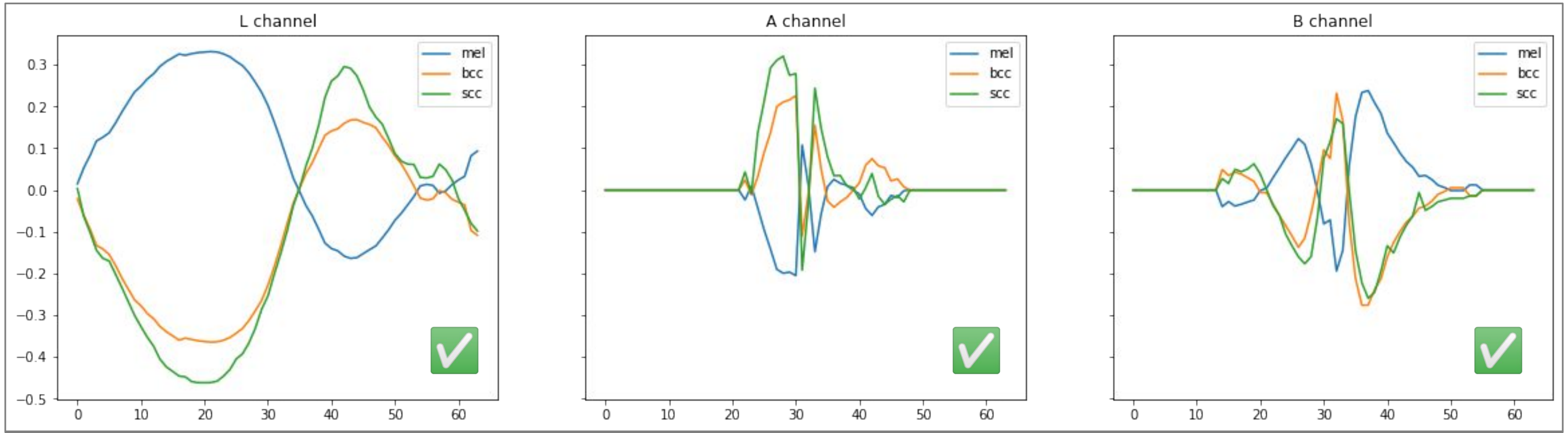



After Normalization

# Mean LAB Histograms For Each Class

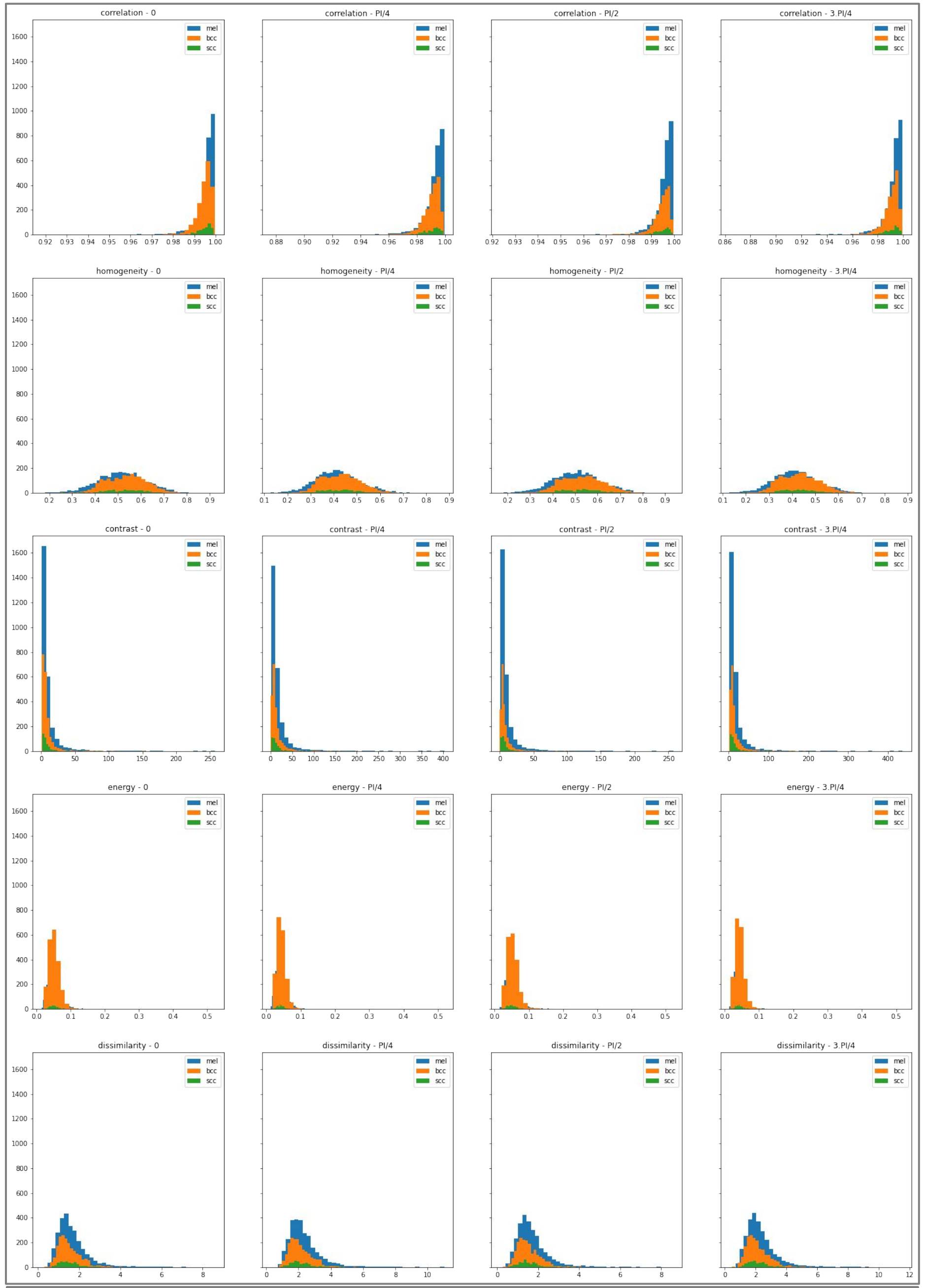



Before Normalization

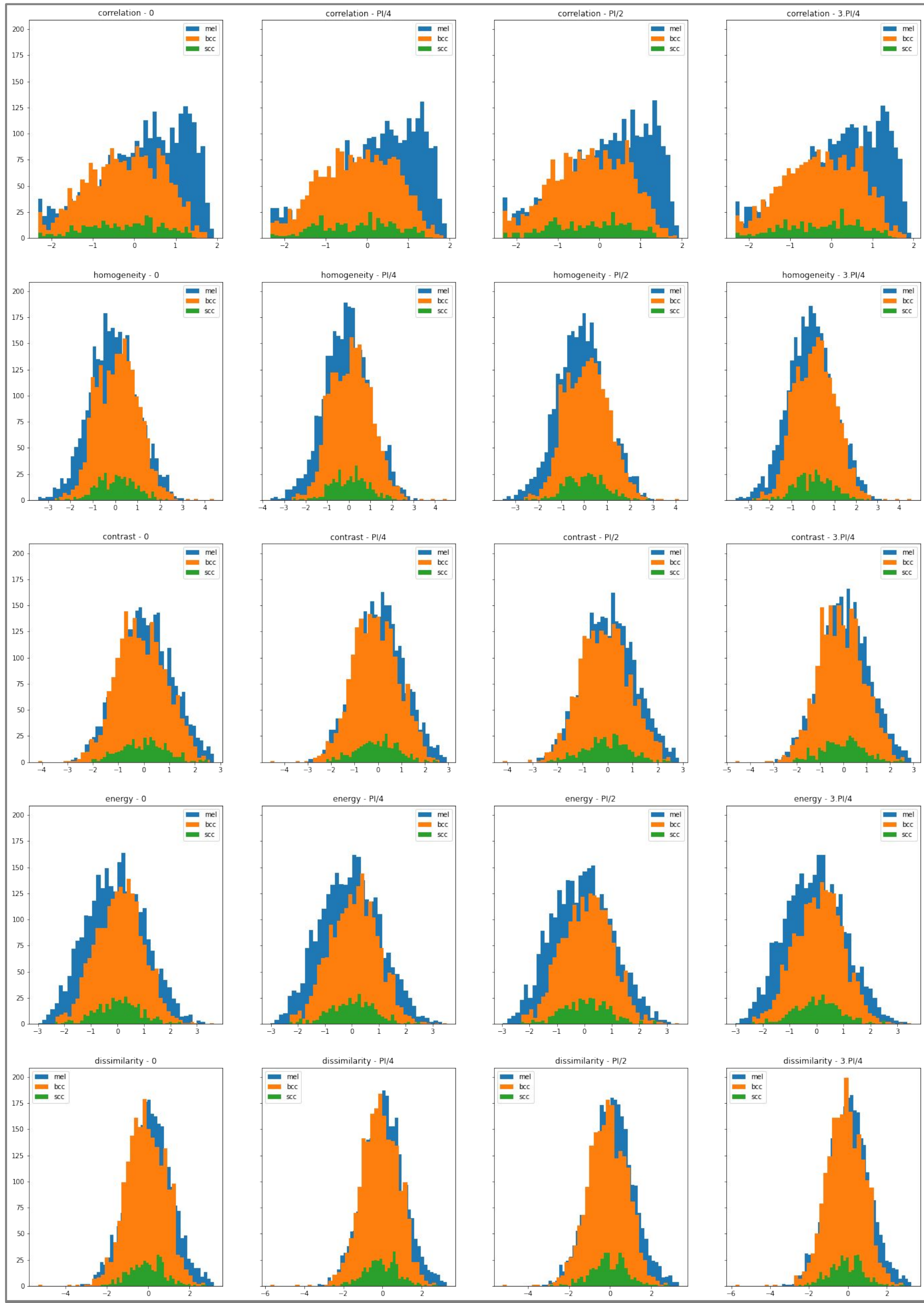



After Normalization

# GLCM Features

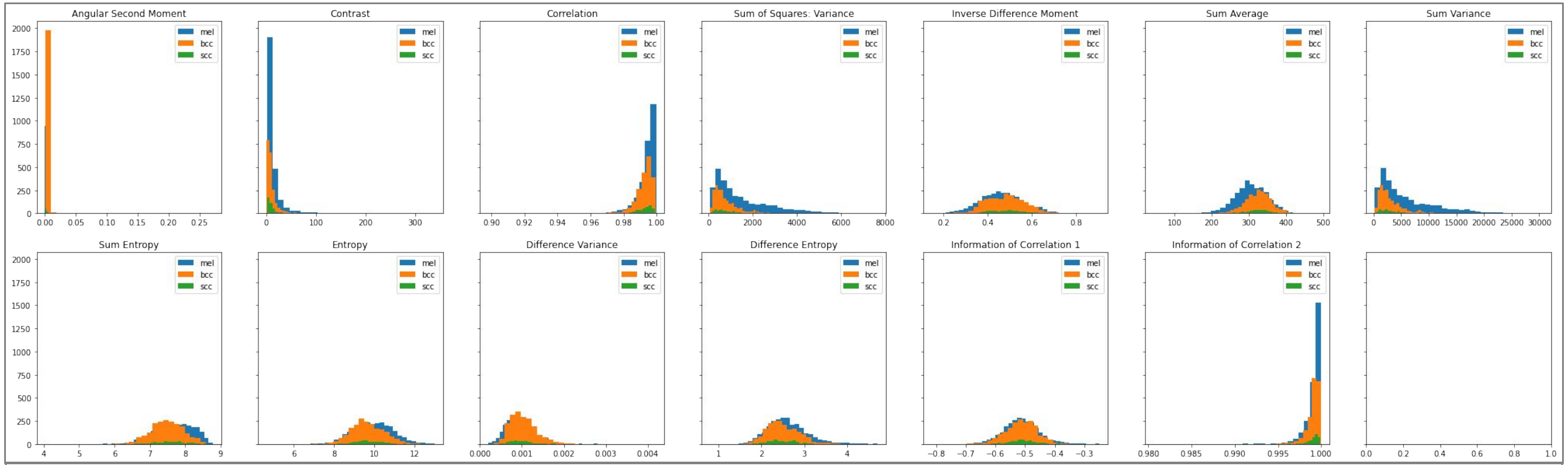


Before Normalization

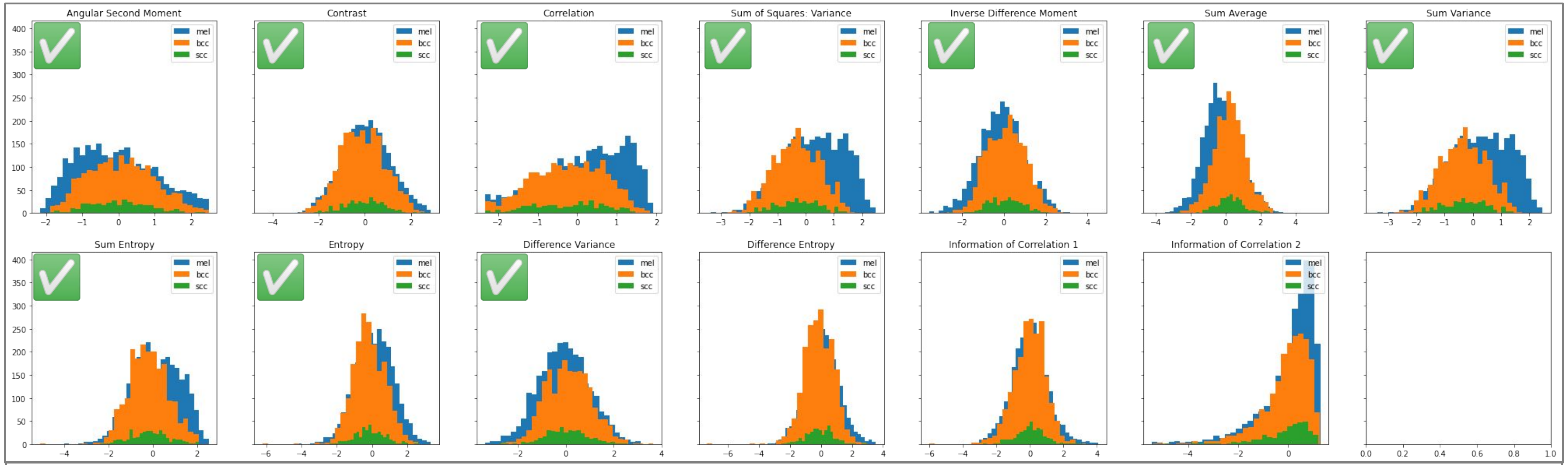


After Normalization

# Haralick Features

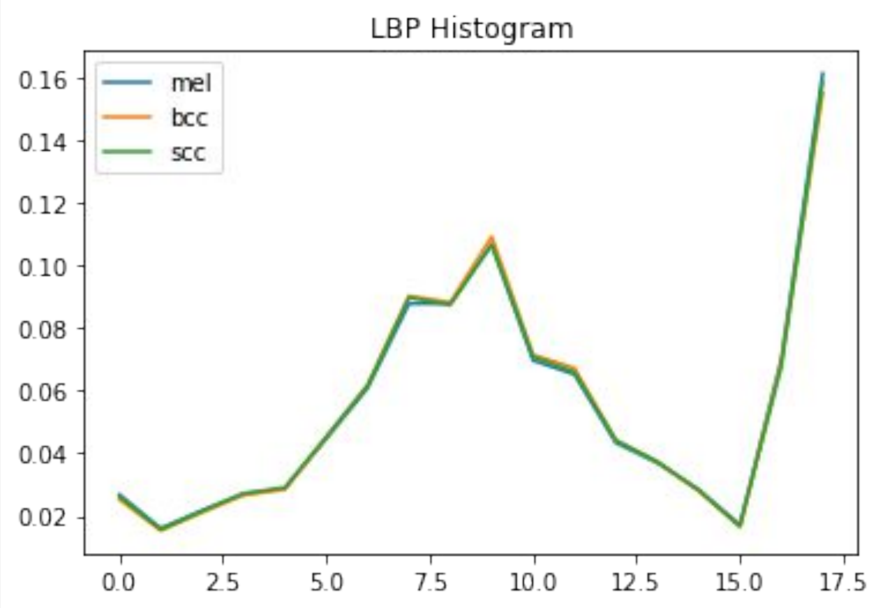



Before Normalization

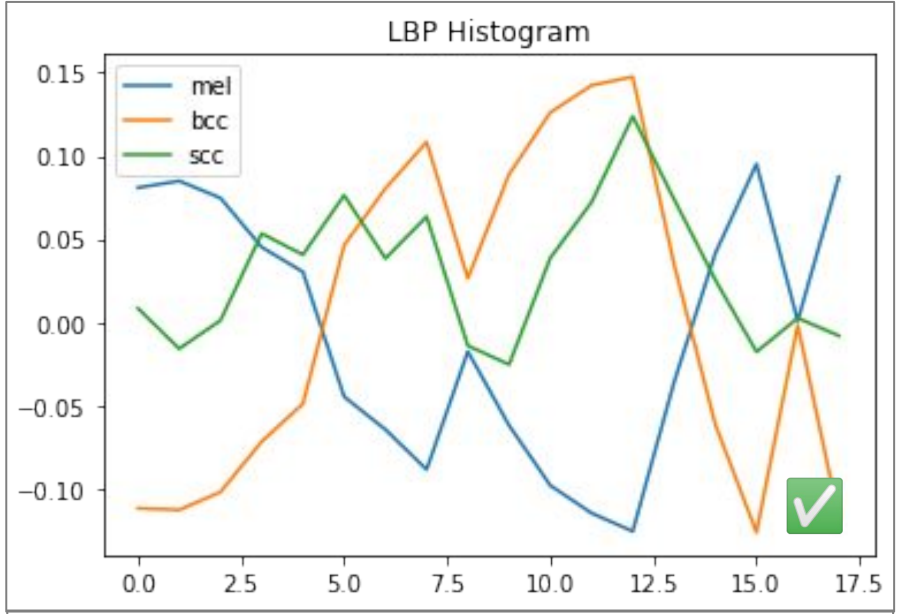



After Normalization

# LBP Histograms




Before Normalization

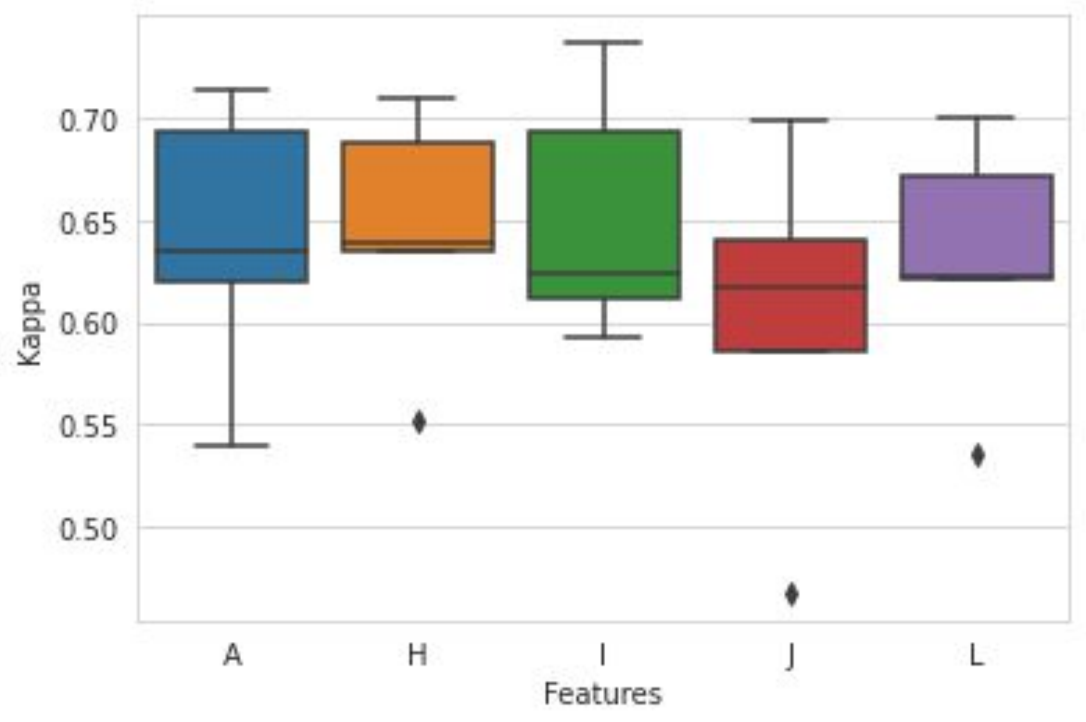



After Normalization

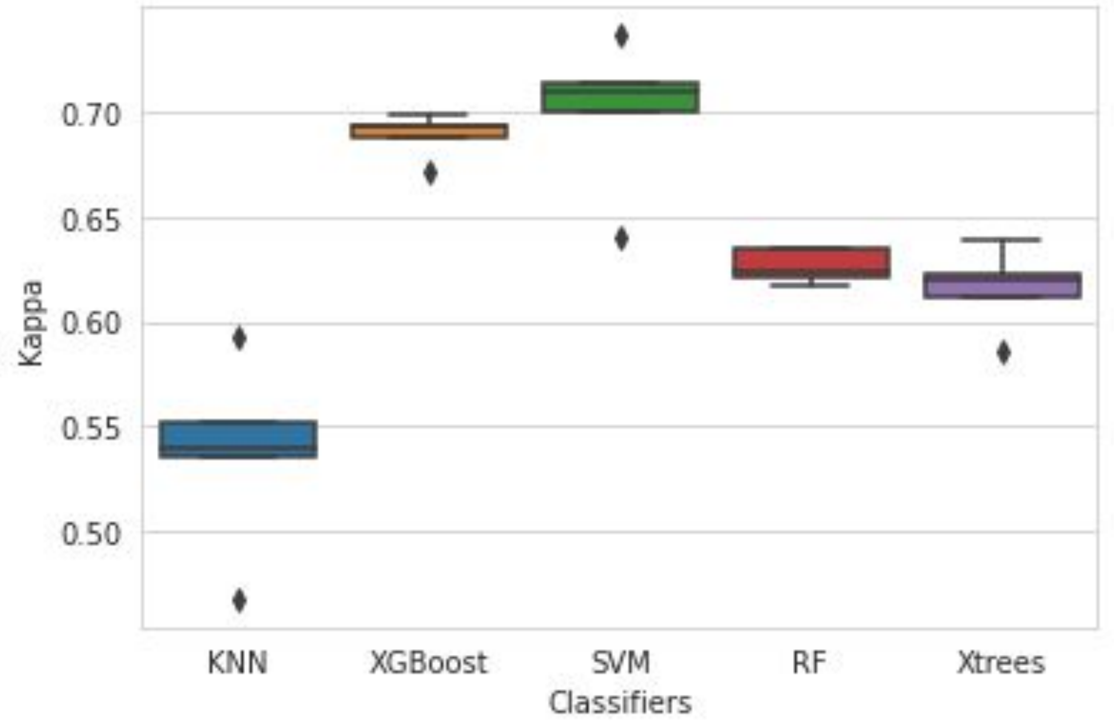
# Training and Evaluation

Feature selection & model training results

# Global Results Evaluation




Boxplot of the kappa metric for each feature vector across all models



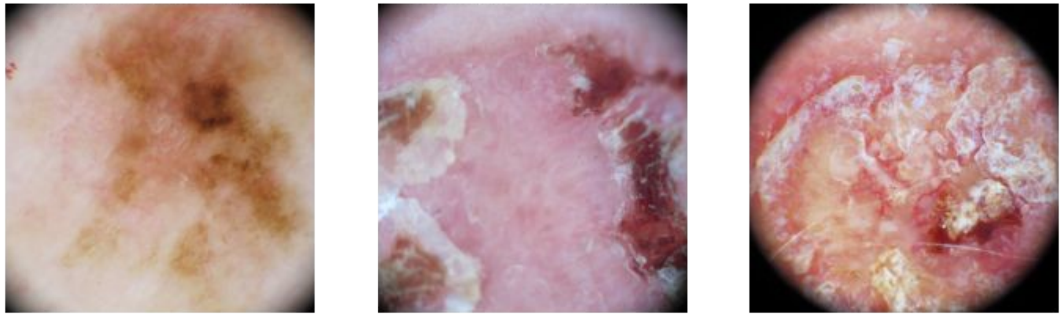

Boxplot of the kappa metric for each classifier across all feature vectors

# Best Results Per Classifier

| Classifier | Parameters | Features | Kappa | Bma | Time (s) |
|---|---|---|---|---|---|
| Baseline NN | k =1 | A | 0.621365 | 0.747204 | 17.2943 |
| K-NN | k=5 | I | 0.593095 | 0.742710 | 14.6484 |
| XGboost | n_estimators = 500, depth = 7, lr = 0.1 | E | 0.693371 | 0.721120 | 419.1023 |
| SVM | C = 11 | I | 0.735959 | 0.747785 | 635.5958 |
| RF | n_estimators = 500, depth = 13, criterion = entropy | A | 0.634628 | 0.698730 | 99.1131 |
| Extra Trees | n_estimators = 1000, depth = 13, criterion = entropy | G | 0.639188 | 0.716365 | 50.6742 |
| Stacking (sklearn) | SVM + XGBoost + XTrees + RF + KNN | A | 0.734990 | 0.741193 | 2300.3030 |
| **Majority Voting (Sklearn)** | **SVM + XGBoost** | **A** | **0.752323** | **0.741561** | **497.1405** |
| Averaging (manual) | SVM + XGBoost + XTrees | A | 0.731742 | 0.742417 | - |
| Hierarchical (manual) | XGBoost | A | 0.652815 | 0.674347 | - |

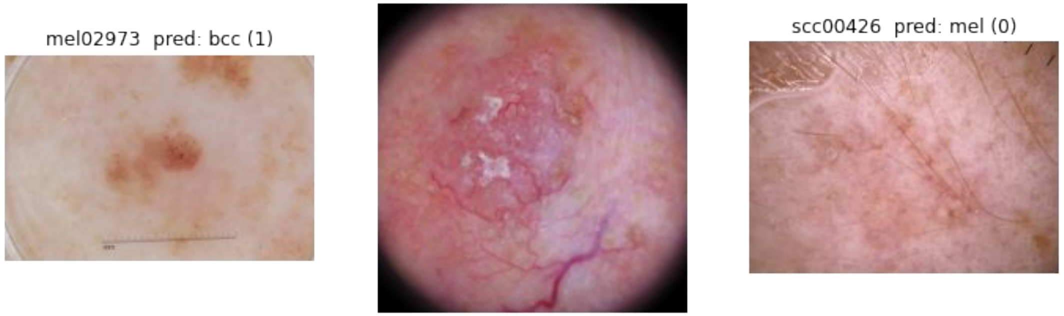
# Best Classifier Results

mel03211 pred: mel (0) | bcc02363 pred: bcc (1) | scc00399 pred: scc (2)

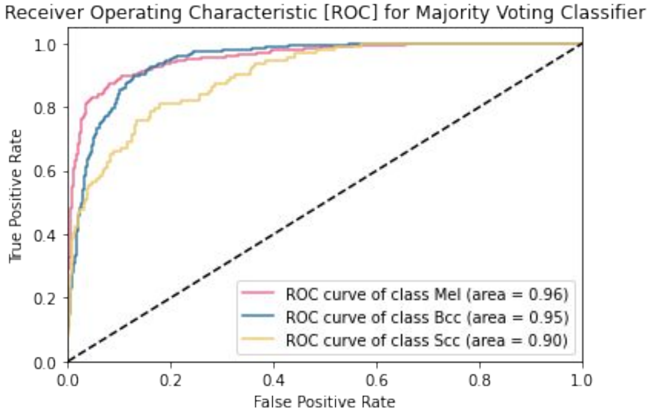


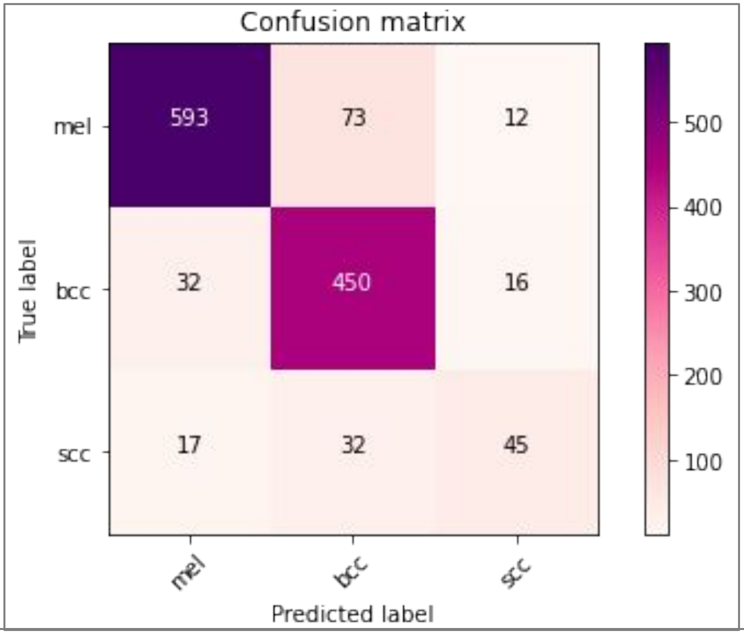
Examples of Correctly Classified Images

mel02973 pred: bcc (1) | bcc02074 pred: scc (2) | scc00426 pred: mel (0)

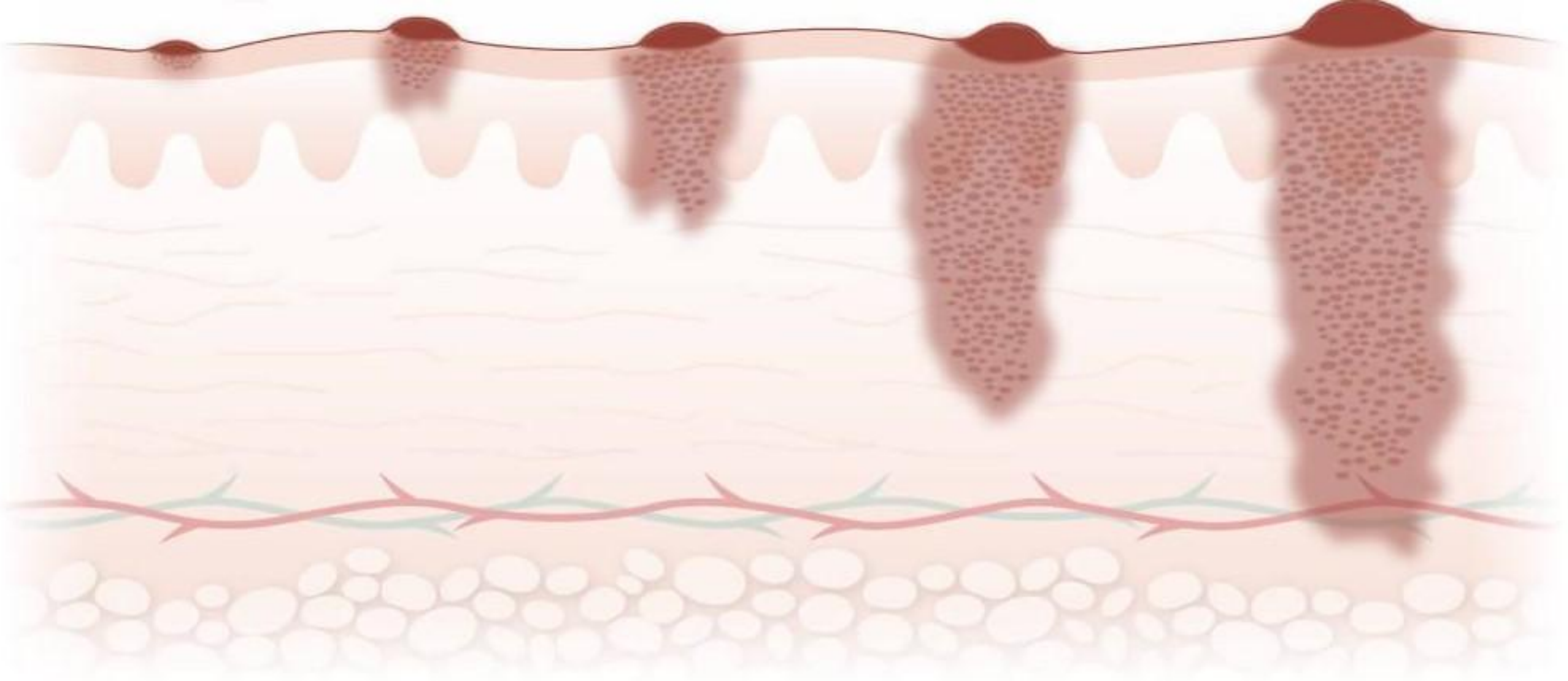


Examples of Wrongly Classified Images



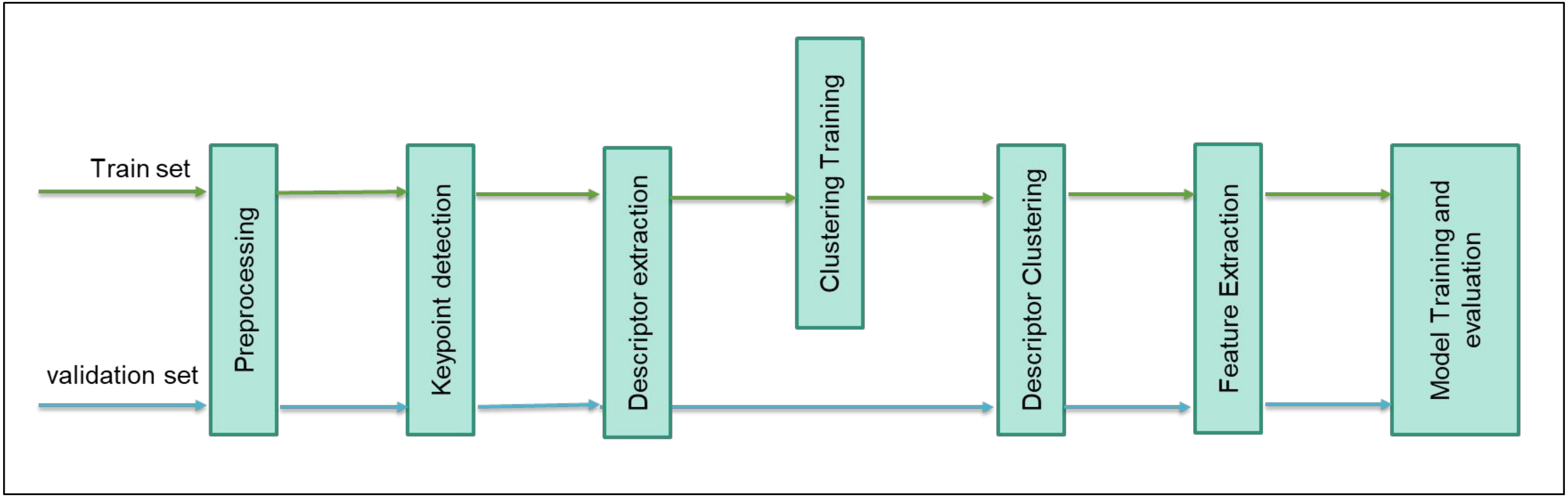




Confusion matrix on the validation set



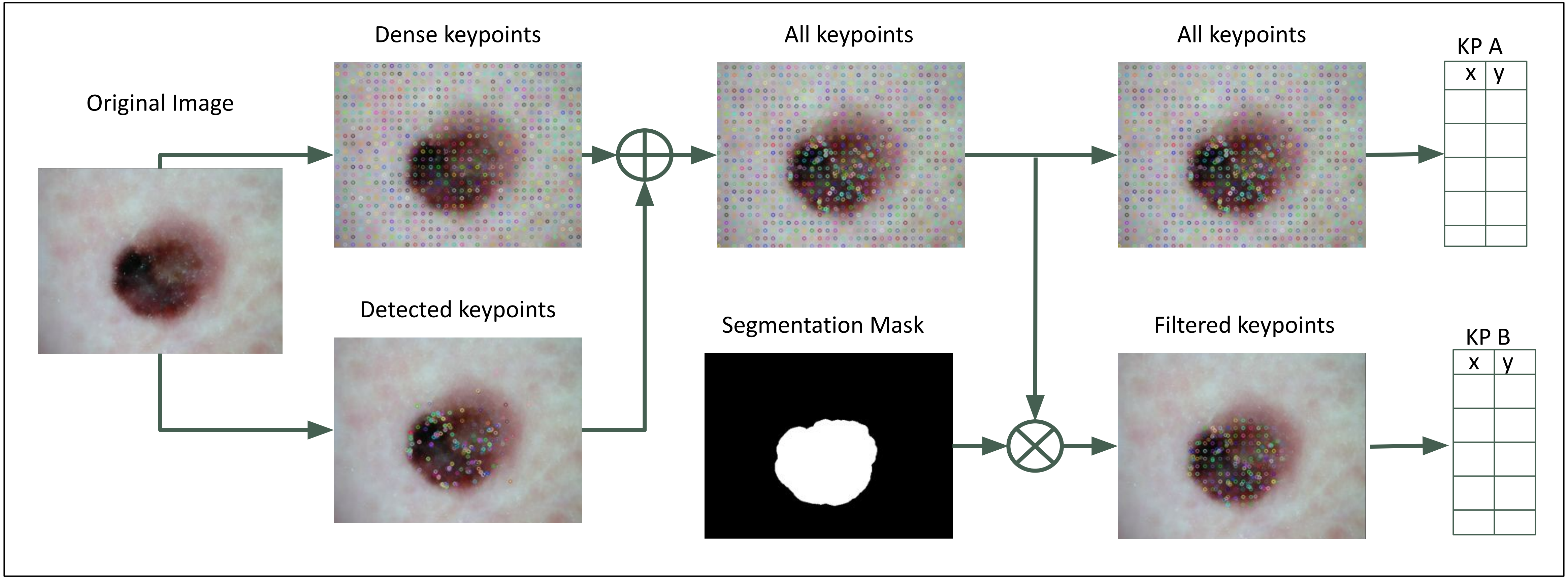
# Bag Of Visual Words

# Pipeline



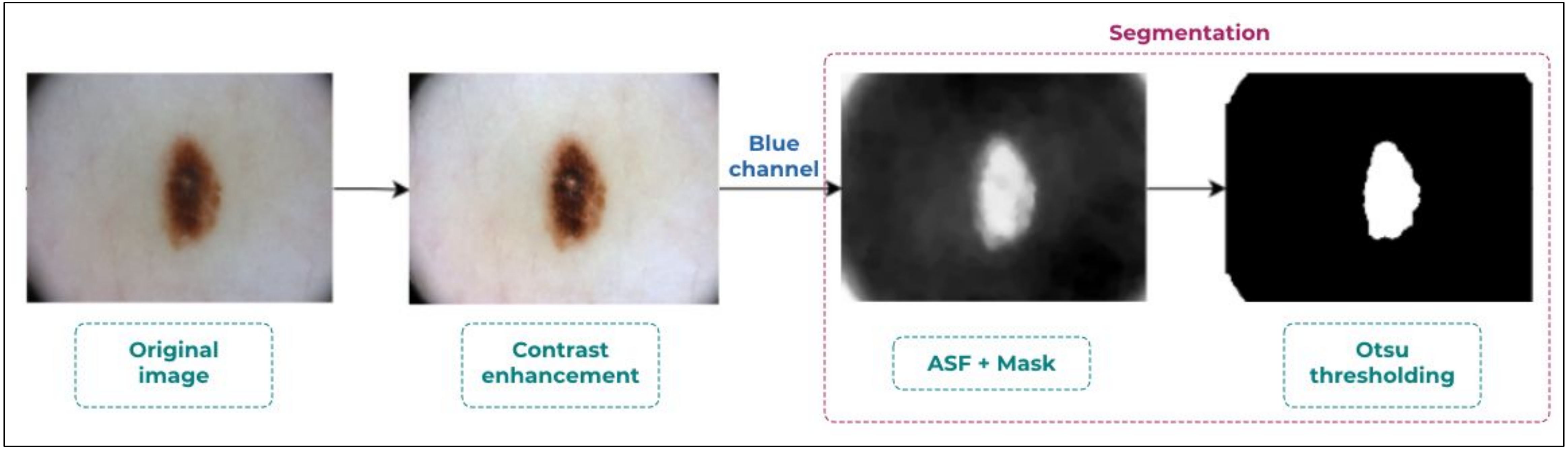

Bag-of-visual-words based classification pipeline

# Keypoint detection




Keypoint Detection Pipeline

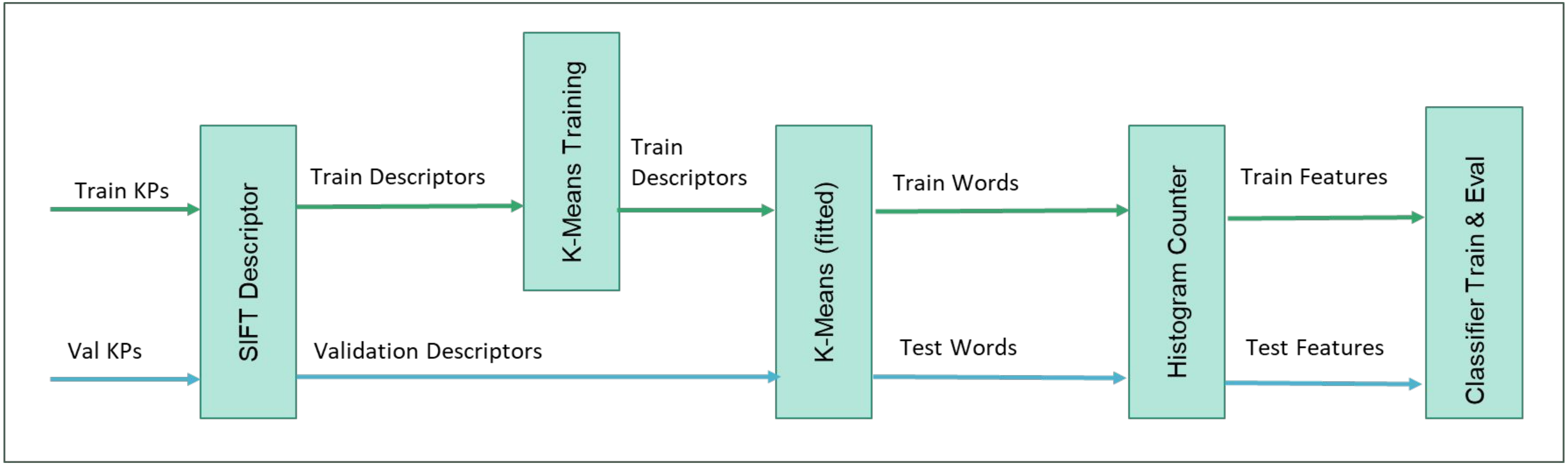
# Segmentation




Segmentation pipeline

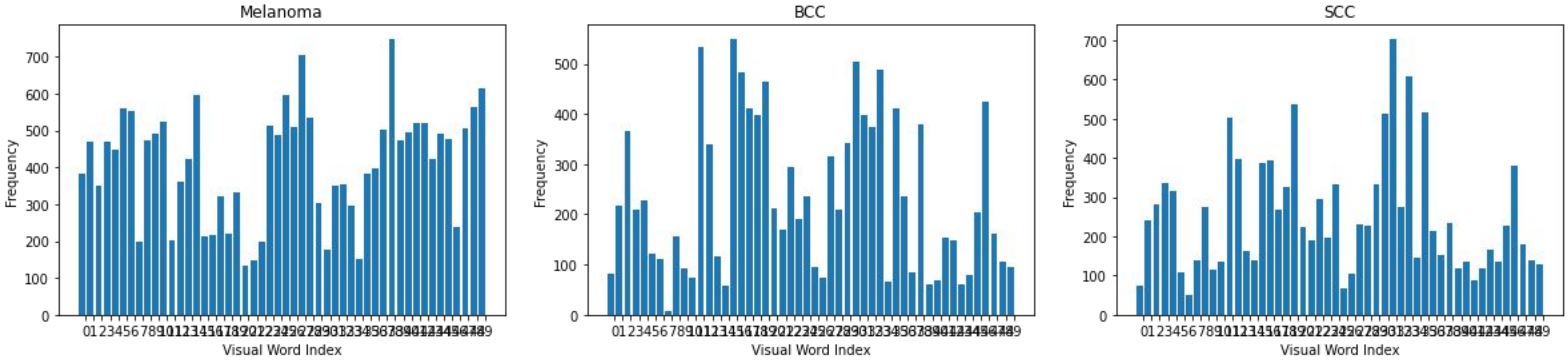
*ASF : Alternating Sequential Filtering to reduce image complexity and make it more homogeneous
*Mask : removing some artefacts (colorful disks) on the borders when using the blue channel

# Feature Extraction




Feature Extraction Pipeline

# Extracted Features




Histogram of the vocabulary occurrences in the train set in each class using SIFT descriptor.

# Model Training and Evaluation

- GridSearchCV with 5 folds was used for parameter estimation.
- SMOTE and undersampling were used for balancing the dataset.
- The following models were evaluated:
  - K-NN
  - SVM
  - XGBoost
  - Random Forest
  - Extra Trees
  - Ensembles of the previously trained models
- The models were compared based on balanced multi-class accuracy and kappa metrics.
- Models were trained using a combination of SIFT + BRISK + ORB features.

# Experimental Results

## With Segmentation mask

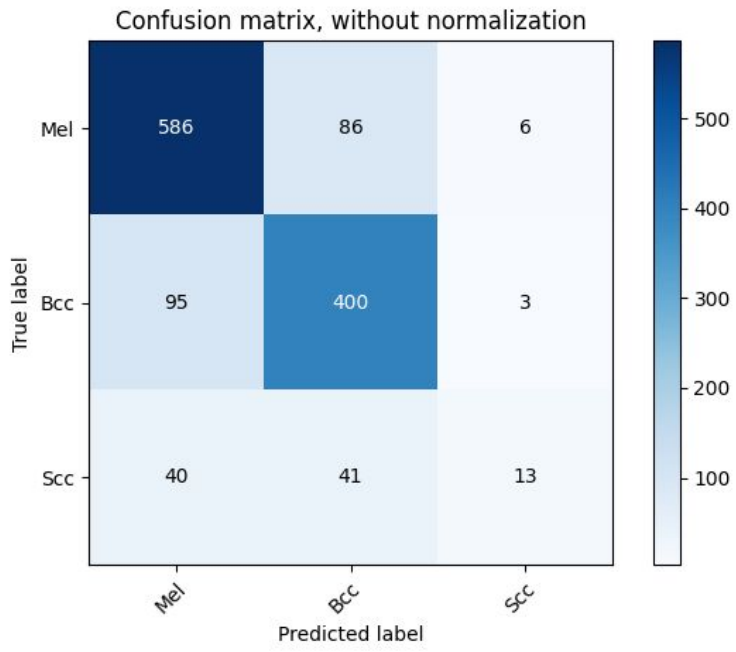
| Model | Train Data | Best Kappa | BMA |
|---|---|---|---|
| NN | All samples | 0.486241 | 0.595114 |
| KNN | All samples | 0.486241 | 0.595114 |
| **XGBST** | **1694 samples/class** | **0.489844** | **0.560988** |
| SVM | All samples + balanced weights | 0.457248 | 0.588034 |
| RF | All samples + balanced weights | 0.478247 | 0.516117 |
| XTrees | 1694 samples/class | 0.472977 | 0.562755 |

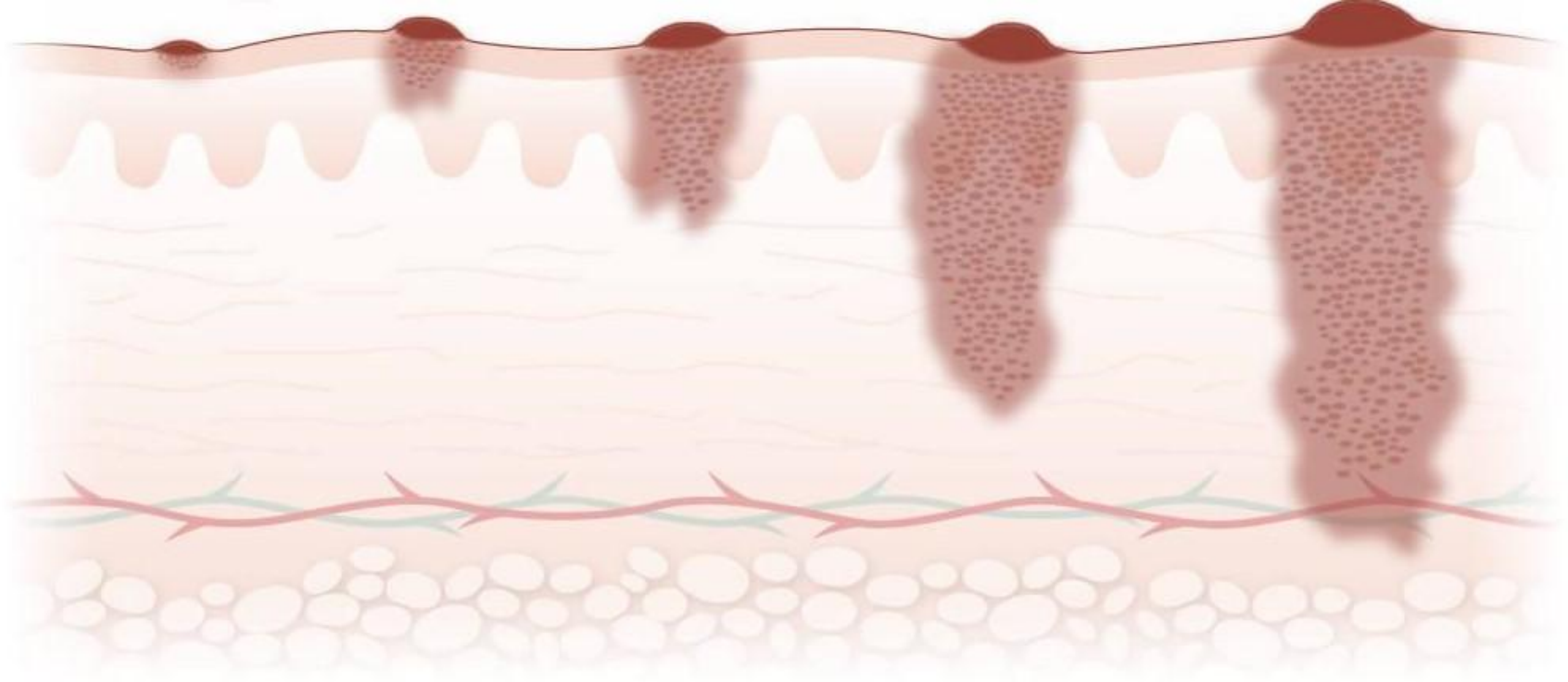
## Without segmentation mask

| Model | Train Data | Best Kappa | BMA |
|---|---|---|---|
| NN | All samples | 0.530192 | 0.646191 |
| KNN | All samples | 0.530192 | 0.646191 |
| **XGBST** | **1694 samples/class** | **0.545448** | **0.601006** |
| SVM | 1694 samples/class | 0.537877 | 0.656429 |
| RF | All samples + balanced weights | 0.515091 | 0.543635 |
| XTrees | All samples | 0.543117 | 0.601907 |

# Experimental Results: SIFT+ORB+BRISK

| Model | Train Data | Best Kappa | BMA |
|---|---|---|---|
| NN | All samples | 0.531296 | 0.661191 |
| KNN | All samples | 0.531296 | 0.661191 |
| XGBST | All samples | 0.579542 | 0.574249 |
| SVM | 1694 samples/class | 0.578807 | 0.637399 |
| RF | All samples | 0.524012 | 0.535706 |
| XTrees | 1694 samples/class | 0.536174 | 0.589971 |
| **Ensemble** | **All data + all models** | **0.599591** | **0.601939** |





# Conclusion

# Conclusion

- For skin lesion classification, Color and texture features play a major role in distinguishing classes.
- Sampling helps tackling imbalanced data classes.
- Normalized features to a gaussian-like distribution minimizes skewness and improves results.
- Ensembling classifiers improves the performance.
- SVM and XGB are indeed robust and they provide the best results.

# Future prospects

- Apply segmentation to enhance the ROI and focus on relevant features only.
- Improve the Hierarchical approach.
- Optimize the BoW approach by tuning the vocabulary size and adding more descriptors (e.g. color)

# Thank you