SIIM-ISIC Melanoma Classification

Introduction:

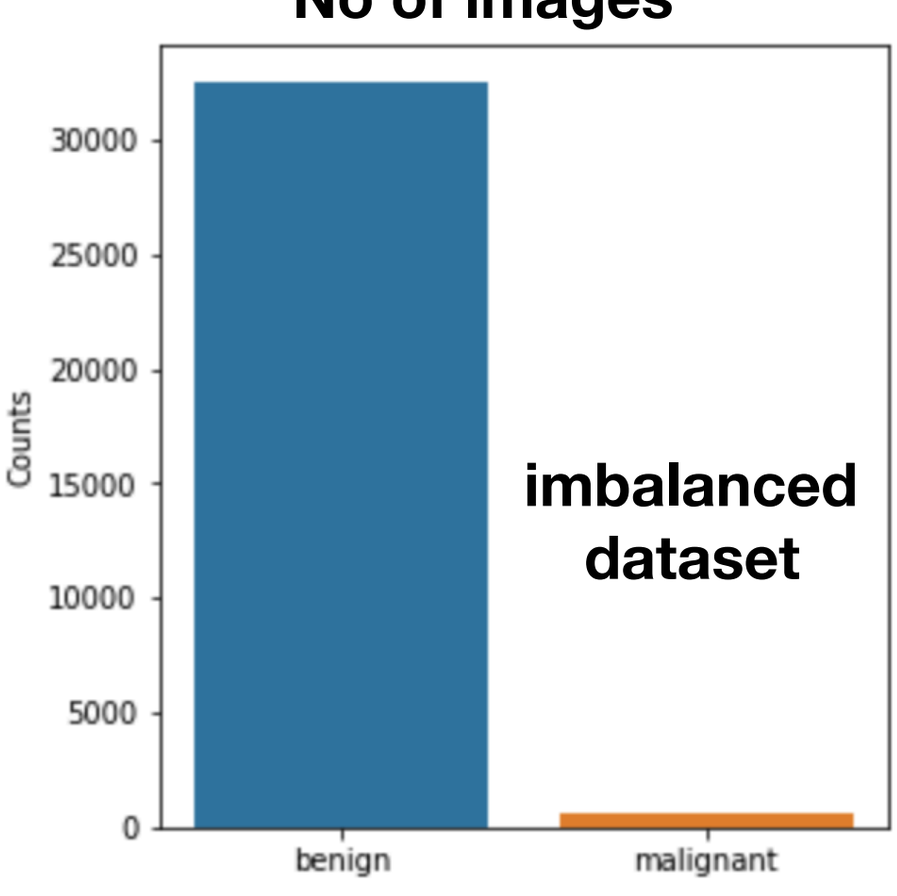
- Dermatologists analyze lesion images to identify melanoma in patient's mole
- Early detection of melanoma can be cured with appropriate treatment
- Efficient computer vision algorithm can reduce visual error and time spent in the manual classification of lesion images
- Society for Imaging Informatics in Medicine (SIIM) pursuit of developing image analysis tools that automate melanoma diagnosis

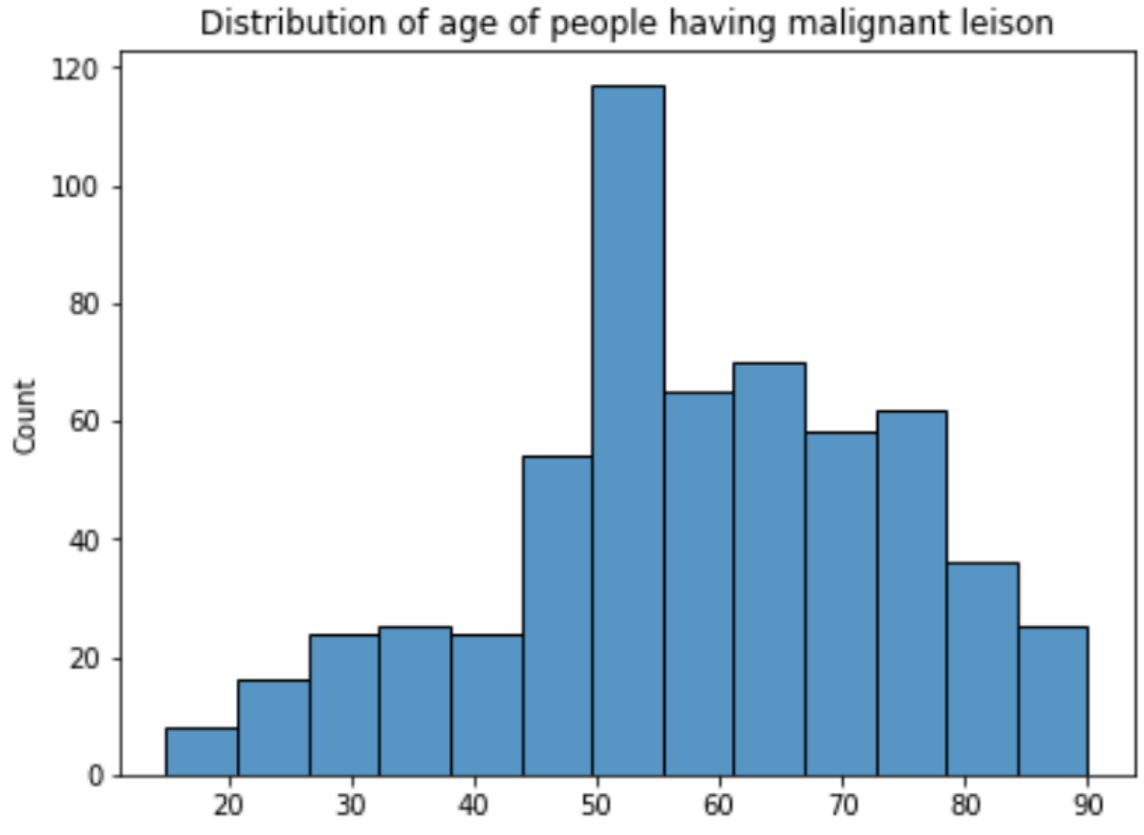
Objective:

- Using ISIC archived dermoscopic images (available in kaggle), develop model to classify lesion images to a benign or malignant class

Dataset

- Kaggle's competition SIIM-ISIC Melanoma classification 2020
- 33,216 images with metadata such as patients id, sex, age, anatomic site, diagnosis, lesion label and target columns referring benign or melanoma
- 32,542 benign images with 584 (1.76%) melanoma images
- 2056 patients whose number of images ranges between 2-115
- Malignant observed across all ages of patient

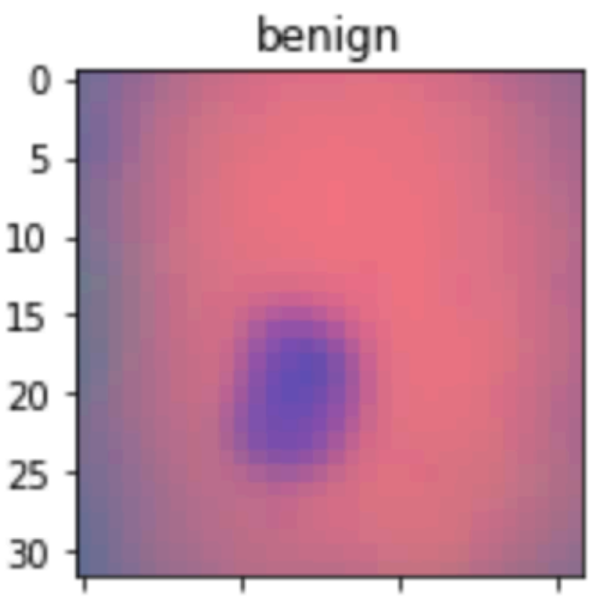


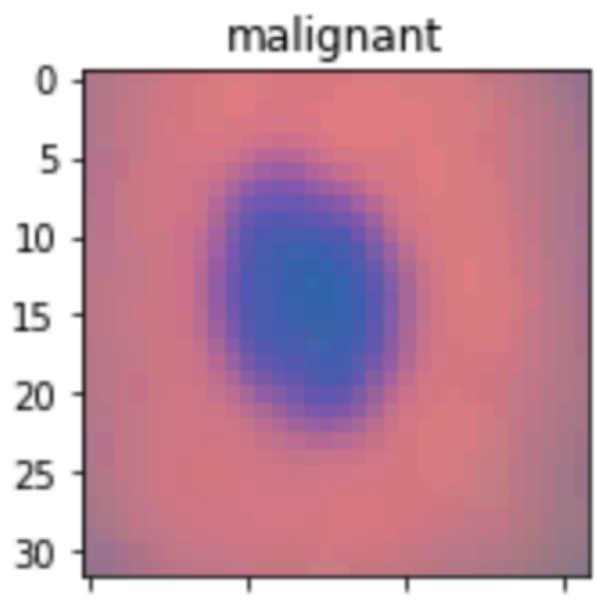


Signs of Melanoma

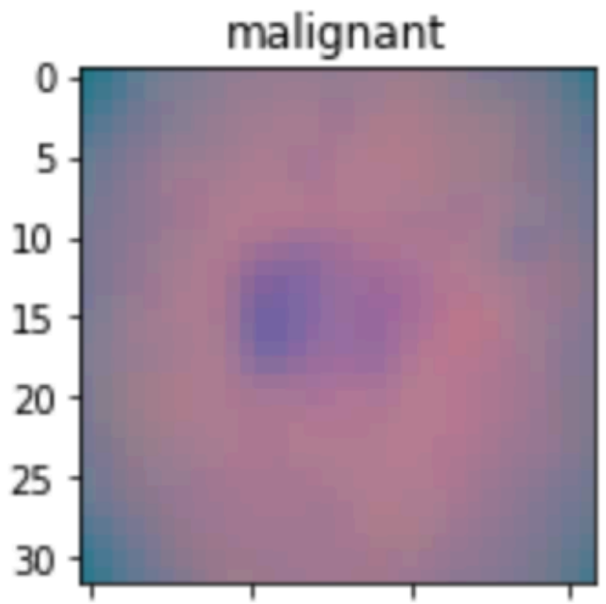
- Asymmetric mole with irregular and uneven edges
- Large brownish spot (diameter >6 mm)
- Mole that changes in color, size or feel or that bleeds
- Lesion that itches or burn

Benign image





Melanoma images



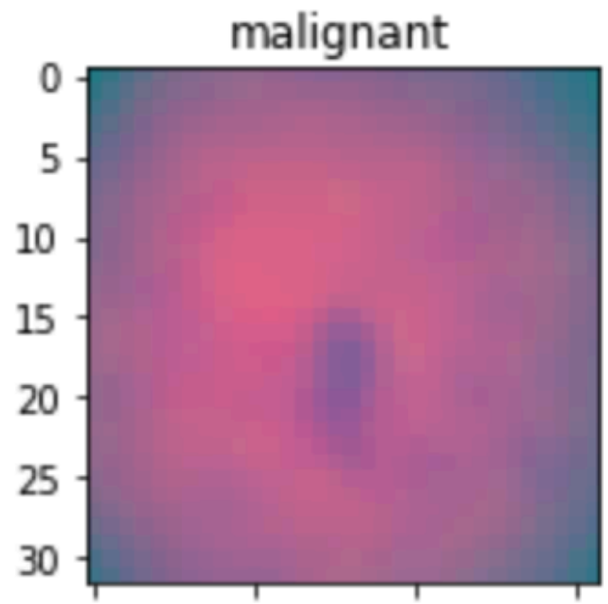


Image Preprocessing

- Image scaling:- resize the images to have fixed height and width
- Color space transformation:- RGB, grayscale
- Contrast enhancement through histogram equalization, adaptive histogram equalization
- Morphology methods:- Gaussian smoothing, threshold, erosion, dilation

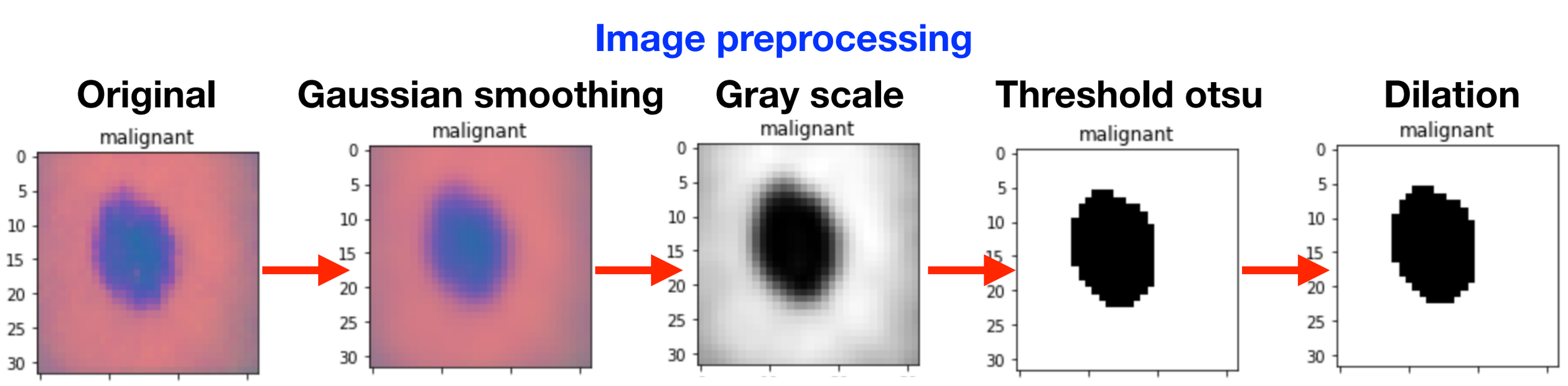
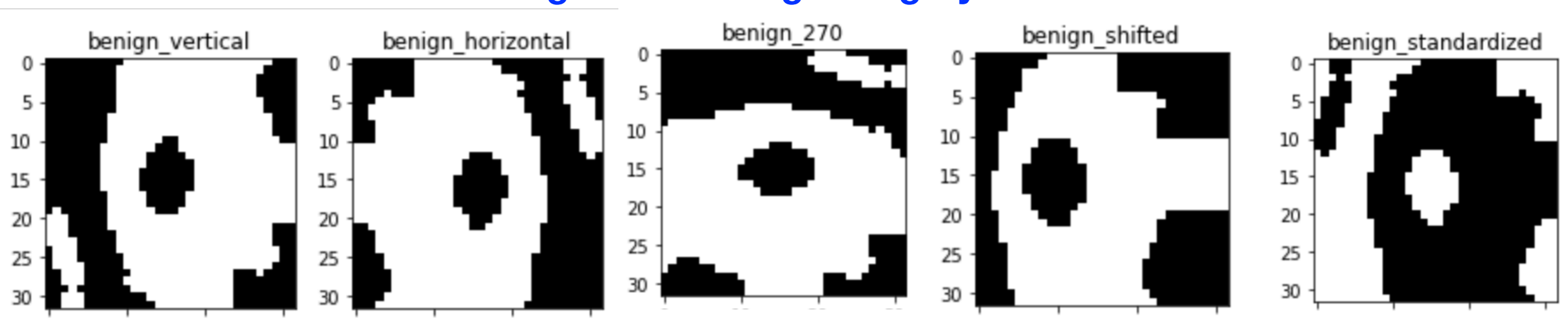


Image augmentation

- Random flip:- Horizontal or Vertical
- Random rotation
- Random shift, shear, zoom
- Random color jitter (brightness, standardize)

Augmented images in grayscale

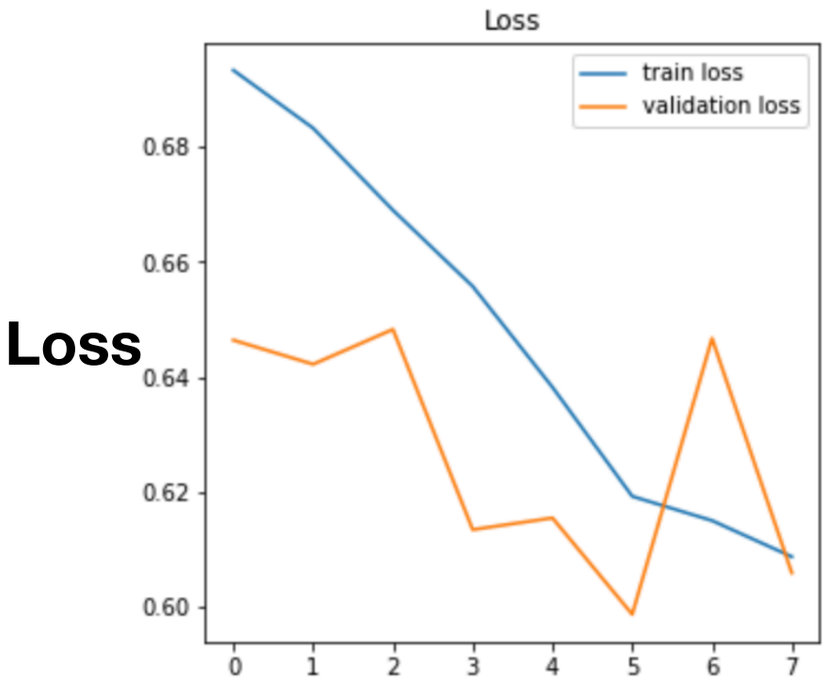


Modeling

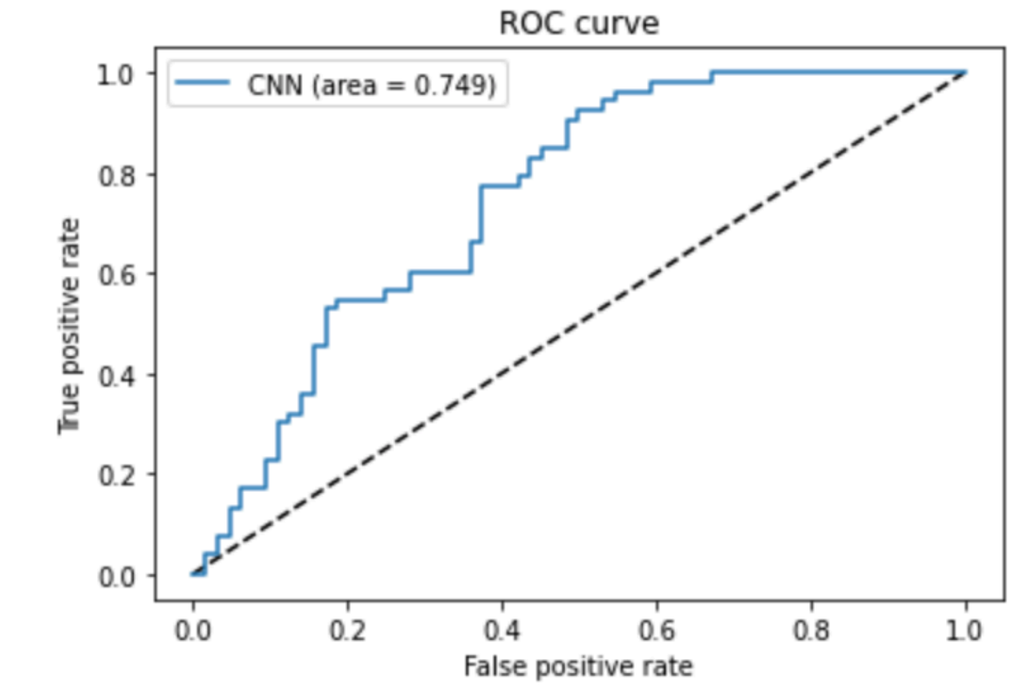
- Built model on equalized dataset
- Image normalization with pixel value between 0 and 1, RGB images with 3 channels
- Deep neural network: four convolutional layers, four max-pooling layers and two dense layers
- Convolutional layers:-
  - First layer: 32-3 x 3 filters
  - Second layer: 64-3 x 3 filters
  - Third layer:- 128-3 x 3 filters
  - Fourth layer- 128-3 x 3 filters
- Max-pooling layers each of the size 2x2 with strides 2
- Optimizer: Adam
- Loss function: Binary crossentropy
- Metrics: AUC
- Regularization with image augmentation, dropout and early stopping

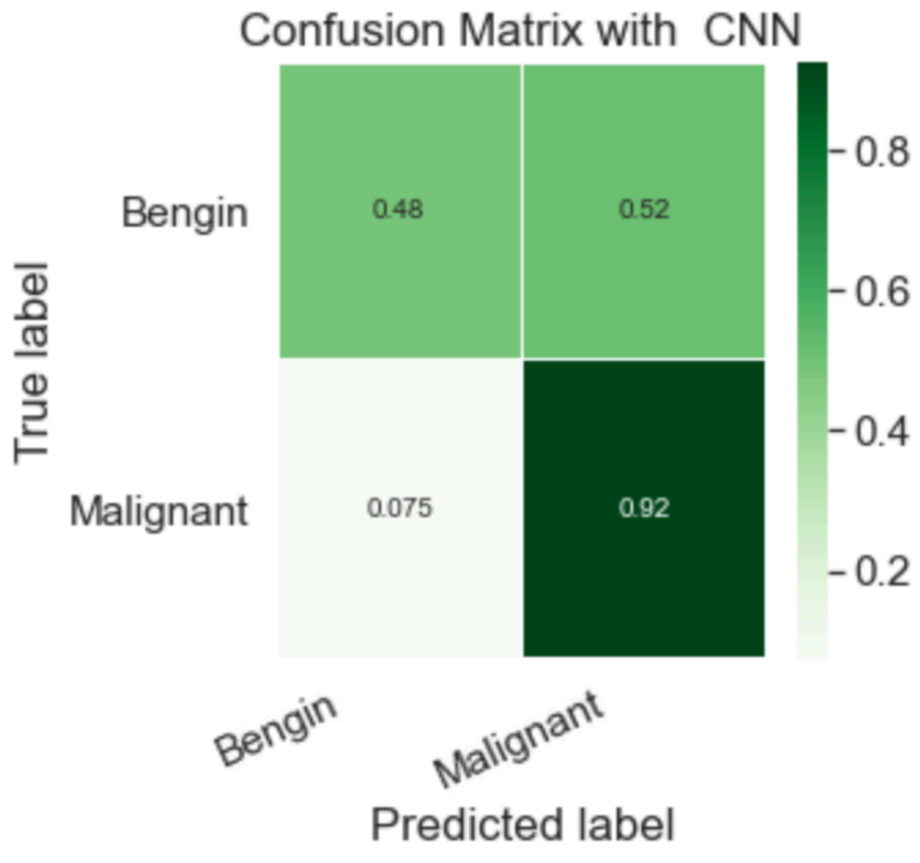
Model evaluation

Loss function with training epochs



- Model predicts malignant images with a probability of 0.92
- Model critical in predicting malignant images than benign images
- The cost of incorrectly classifying malignant images are much risker than incorrectly classifying benign images



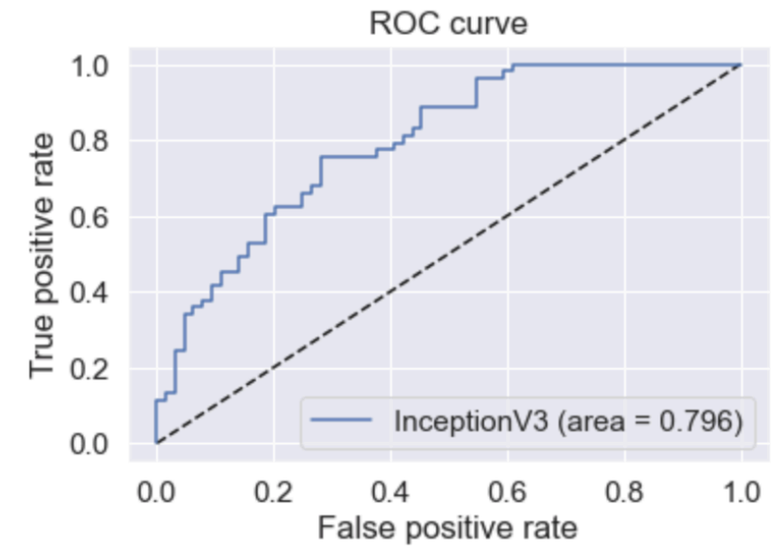


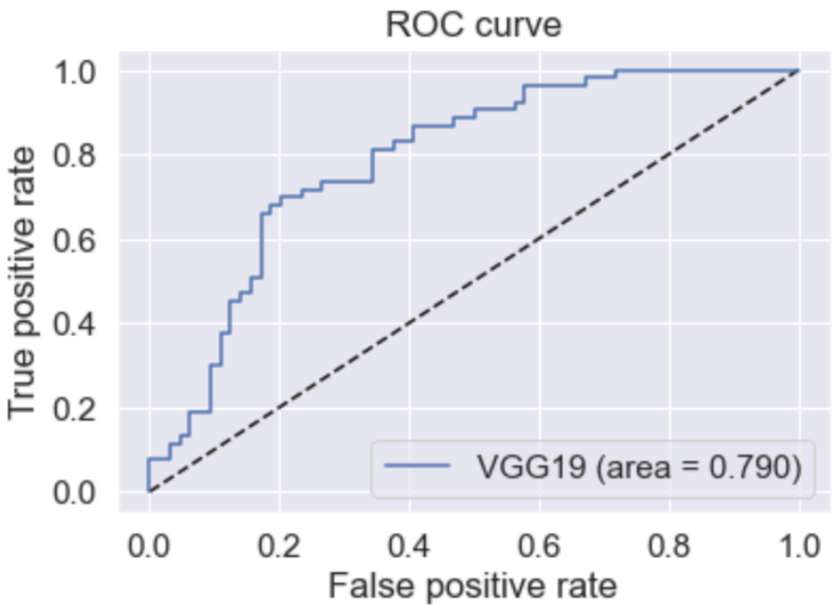
Transfer learning

- Build models with weights trained on imageNet and EfficientNet
- Start the learning process from the patterns that have already been learned
- Employed models
  - VGG19
  - Resnet50 and Resnet152
  - InceptionV3
  - EfficientNetB0 and EfficientNetB5
- Trained model fixing all layers or allowing few top layers to be trained

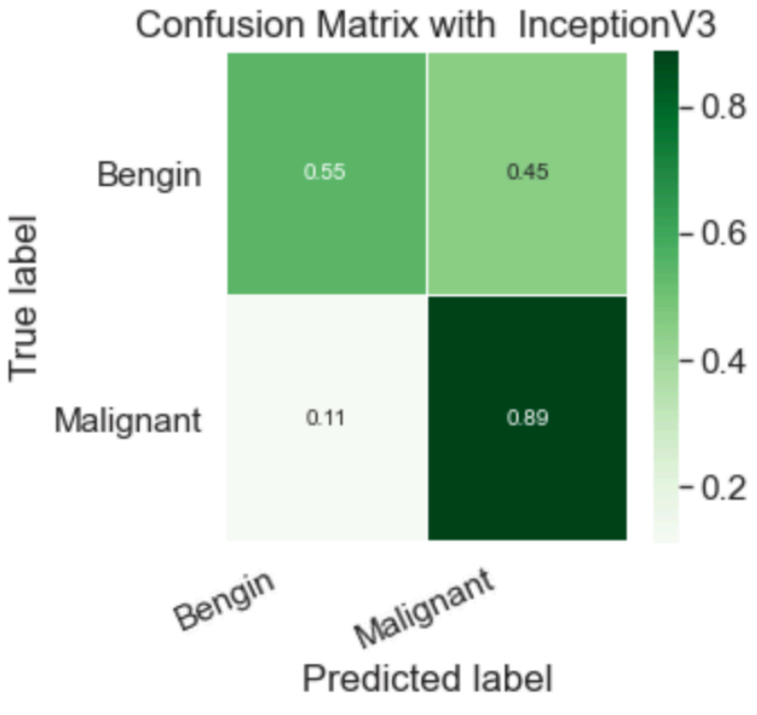
Model evaluation

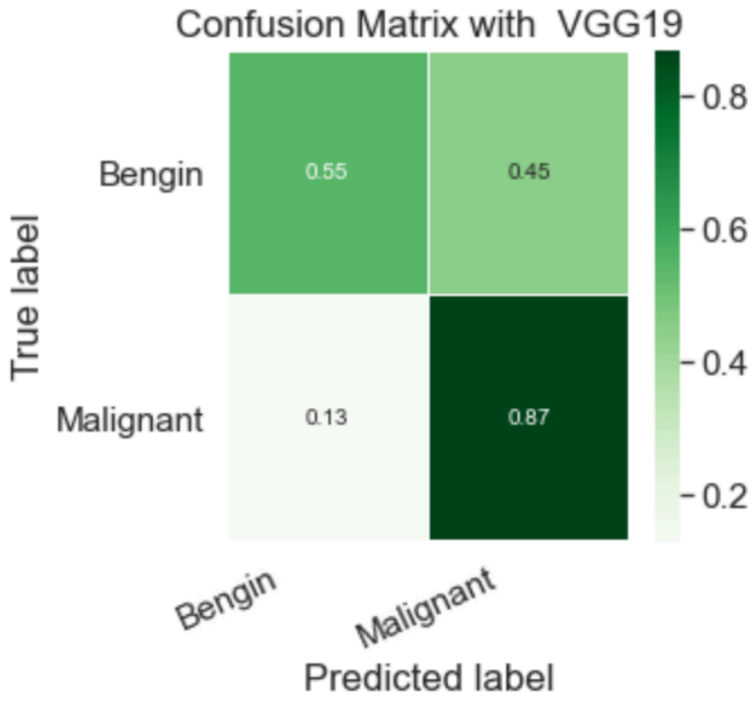
Model	Loss	AUC	Precision	Recall	F-score
CNN	0.59	0.75	0.48	0.92	0.63
VGG19	0.63	0.79	0.55	0.87	0.67
Resnet50 and 152	0.71	0.70	0	1	0
Inception V3	0.60	0.80	0.55	0.89	0.68
EfficientNetB0 and	0.695	0.39	0	1	0





-





Conclusions

- **Developed binary models that can identify images to a benign or malignant class**
- **Multiple models were built using deep neural networks and transfer learning architecture**
- **CNN achieved precision of 0.48, recall of 0.92, F1 score of 0.63 and AUC of 0.75**
- **VGG19 and inceptionV3 model performed better than CNN on precision, F1 score and AUC**
- **Resnet and EfficientNet couldn’t performed better than CNN**

Recommendations

- Develop model on imbalanced dataset
- Fine tune CNN with hyper parameters such as batch size, learning rate, average pooling, filter size and strides
- Evaluate model performance with other pre-trained models such as Xception, VGG16, MobileNet, Densenet, NASNet, MobileNetV2
- Fine tune pre-trained models freezing all layers or making few layers trainable