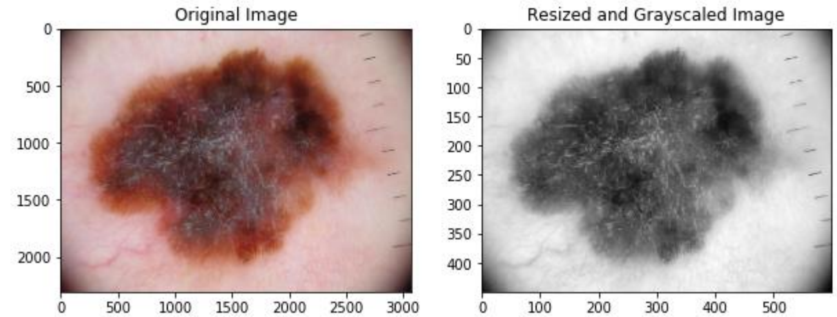
PROJECT DESCRIPTION AND SUMMARY

PROJECT GOAL

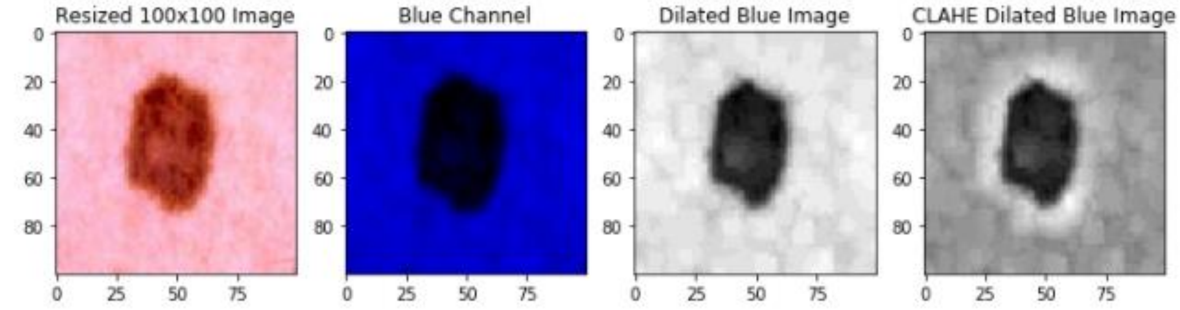
This project compares the pixel-based approach of extracting image features of dermoscopic images of skin lesions and novel approaches using feature engineering and image de-noising techniques coupled with other feature extraction methods to classify the images as either malignant or benign. This would help prove improvement in image classification and help guide decisions towards inclusion of these methods in clinical disease detection systems.

PROJECT APPROACH

From the first approach, the minimum image dimension was determined from the image set and was used to rescale the other images to 450x600 pixels and grayscale in order to reduce the number of channels. Features related with images using pixel value extraction by creating an array on image pixel values on a range of 0 – 255. The features of skin images were reduced using principal component analysis to the more essential features. In the classification stage, three classifiers based on supervised machine learning were developed namely K-Nearest Neigbour, Logistic Regression and SVM.



Histogram of Gradients (HOG) was used for the second approach to extract the gradient and orientation of the image edges in order to describe the image features. A number of Image pre-processing techniques were considered to minimize Image noise such as ruler markings, hair, air bubbles, poor lighting among others. They include resizing the images to 100x100 pixels, using only the blue channel for the images, morphological closing by using dilation to segment the lesions from the skin and Contrast Limited Adaptive Histogram Equalization (CLAHE) balance the image lighting and classifying using Gaussian Naïve Bayes and SVM.



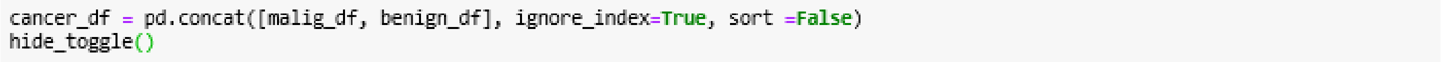
RESULTS

The models were trained and tested in both approaches with their hyperparameters tuned manually and cross-validated. In the first approach, K-NN performed best with an accuracy of 63.3% and cross validation score of 59.6% while in the second approach Gaussian Naive Bayes performed best with an accuracy of 67.7% and cross-validation score of 75.7% which is a clear improvement over the previous method.

DATA PROCESSING FOR GOAL 1

Exploratory data analysis (EDA) was performed on the benign and malignant images to determine their maximum and minimum dimension size of (4459,6688) and (450,600) respectively for both classes of images. All the images from dataset were then scaled down to 450x600 and in order to reduce computational cost and training time, the image features were also scaled down by converting to grayscale.

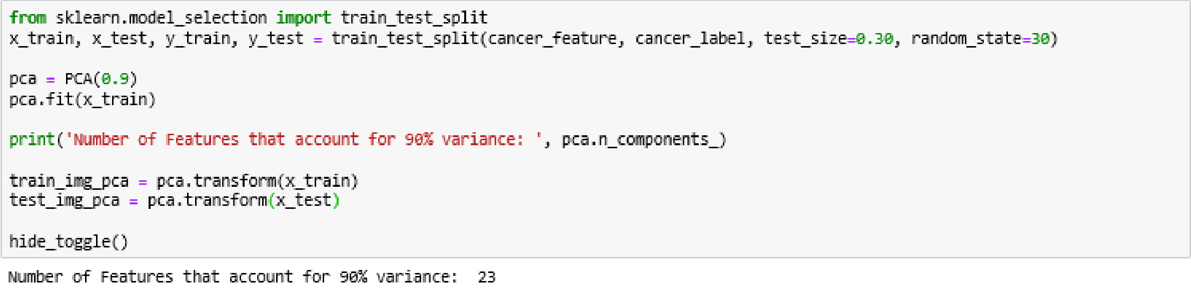
The pixel values from the converted images were then extracted into an array and thereafter transformed into a dataframe with the class of each attached as the final column. The malignant class was labelled 0 while the benign class was labelled 1. This process was done for each class and concatenated to one dataframe containing 27000 features and 300 instances.



```
cancer_df = pd.concat([malig_df, benign_df], ignore_index=True, sort =False)
hide_toggle()
```

Since the number features in the dataframe was large, dimensional reduction would have to be performed and in order to prepare for this. Hence, the dataframe was scaled normally and classed.

This dataframe was then randomly split into a training and test set in the ratio 70/30 respectively with PCA applied to the feature train set in order to determine the features that accounted for 90% of the variance. It turned out that 23 features were responsible for this out of 27000 features. This was then used to transform the feature train and test.



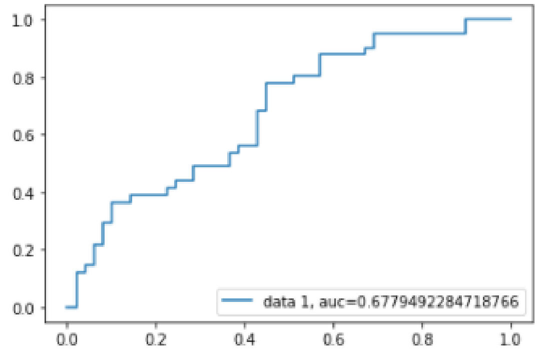
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(cancer_feature, cancer_label, test_size=0.30, random_state=30)

pca = PCA(0.9)
pca.fit(x_train)

print('Number of Features that account for 90% variance: ', pca.n_components_)

train_img_pca = pca.transform(x_train)
test_img_pca = pca.transform(x_test)

hide_toggle()
```

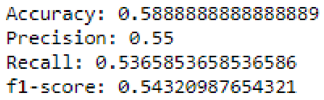
```
Number of Features that account for 90% variance:  23
```

CLASSIFICATION MODELS BASED ON PIXELS

After processing the image features, the next step was to determine classifier algorithms which could properly classify the test image features after fitting on the train features. Three classifiers were then chosen; Logistic Regression, Support Vector Machines and K- Nearest Neighbour.

Logistic Regression was the first base model and it was fitted with the transformed train features and the train class. The performance on the test set was measured on accuracy, precision, recall and f-1 score. Its hyper-parameters were manually tuned with C set to 1 after several predictions. Its performance was 0.589 accuracy, 0.55 precision, 0.537 recall and 0.54 f-1 score.

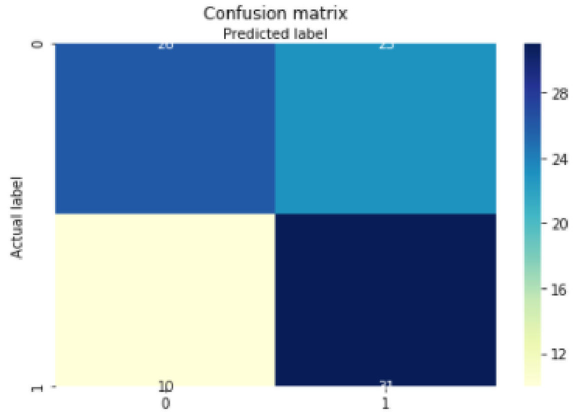




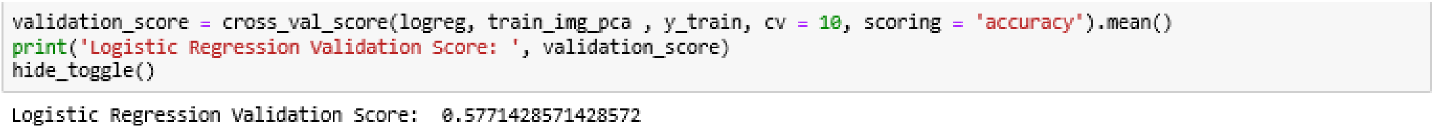
```
Accuracy: 0.5888888888888889
Precision: 0.55
Recall: 0.5365853658536586
f1-score: 0.54320987654321
```

Its area under curve (AUC) was also plotted to understand its predictions with a score of 0.68.

A confusion matrix was also plotted to understand the logistic regression predictions vs the actual class on whether it predicted correctly or falsely. It correctly predicted 31 instances as malignant and 22 instances benign while it incorrectly classified 18 malignant instances as benign and 19 benign instances as malignant.



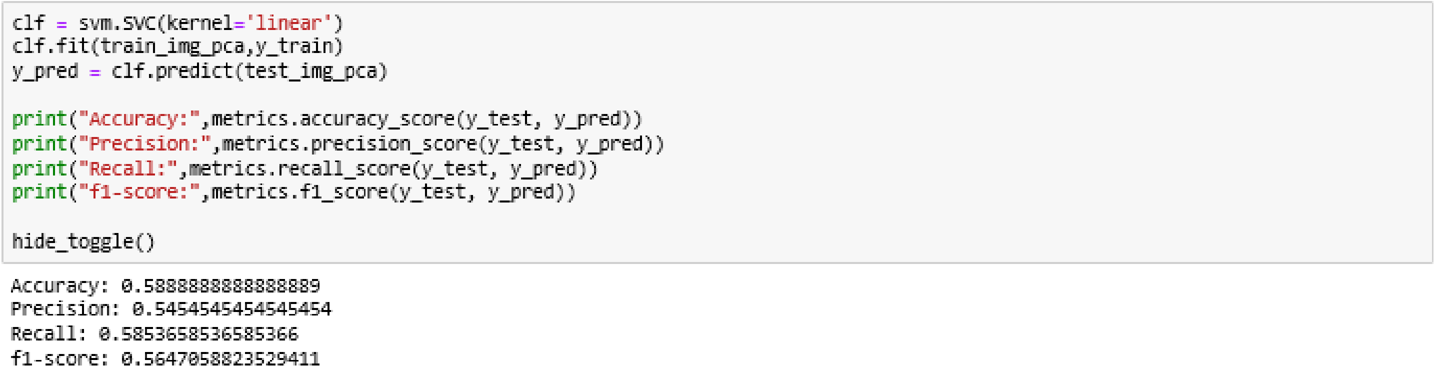
The results were then cross-validated where the logistic regression model made predictions on the training set iteratively 10 times. The mean score of the iterations was then evaluated as 0.577.



```
validation_score = cross_val_score(logreg, train_img_pca , y_train, cv = 10, scoring = 'accuracy').mean()
print('Logistic Regression Validation Score: ', validation_score)
hide_toggle()
```

```
Logistic Regression Validation Score:  0.5771428571428572
```

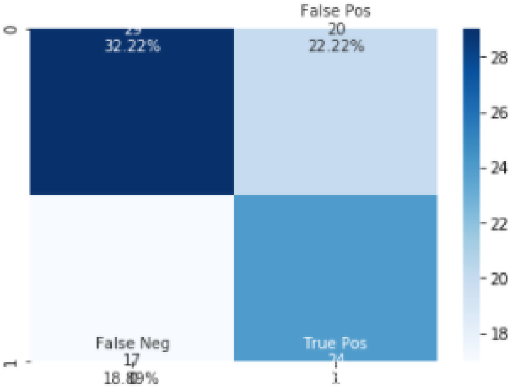
SVM was considered after and it was also fitted with the train set to predict the test set. Its hyperparameter was also manually tuned where its kernel was set to linear. It obtained an accuracy of 0.589, precision of 0.55, recall of 0.59 and f-1 score of 0.56 which were similar to what the logistic regression obtained.



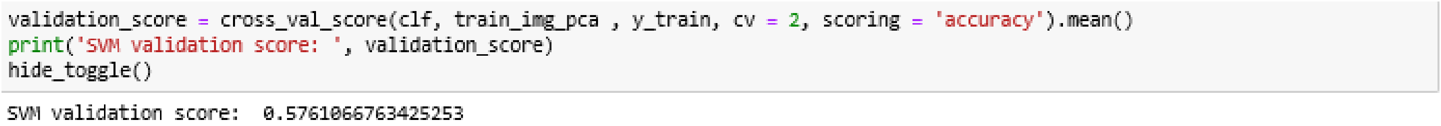
```
clf = svm.SVC(kernel='linear')
clf.fit(train_img_pca,y_train)
y_pred = clf.predict(test_img_pca)

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
print("f1-score:",metrics.f1_score(y_test, y_pred))

hide_toggle()
```

```
Accuracy: 0.5888888888888889
Precision: 0.5454545454545454
Recall: 0.5853658536585366
f1-score: 0.5647058823529411
```

According to the confusion matrix for SVM, it predicted malignant correctly 29 times and benign correctly 24 times. It misclassified benign as malignant 20 times and vice versa 17 times.



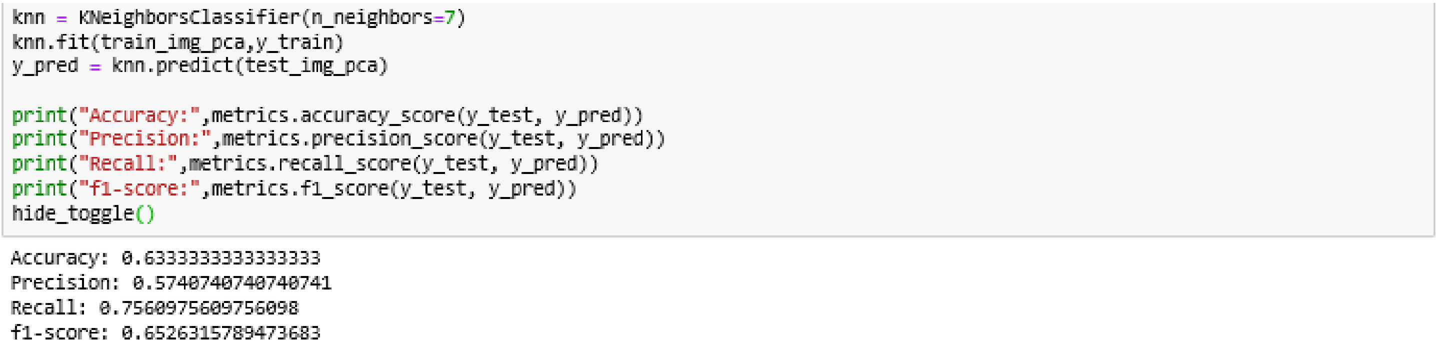
The SVM model was cross validated twice due to its computation time. It achieved a score of 0.576 which was also similar to the logistic regression.



```
validation_score = cross_val_score(clf, train_img_pca , y_train, cv = 2, scoring = 'accuracy').mean()
print('SVM validation score: ', validation_score)
hide_toggle()
```

```
SVM validation score:  0.5761066763425253
```

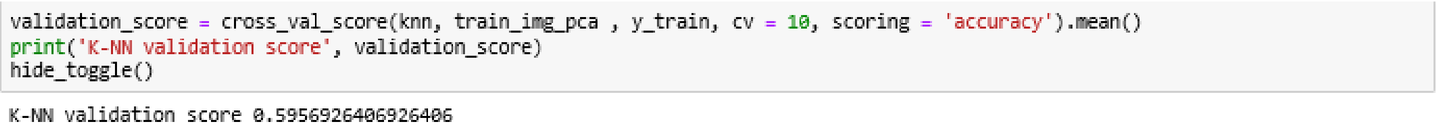
The final classifier considered for the pixel-based approach was K nearest neighbor. It was expected to perform better due to its non-parametric and lazy method of classification which assumes no underlying distribution and the small number of features. It was fitted with its number of neighbors hyperparameter set to 7 by empirical deduction. Its accuracy was evaluated to 0.633, precision at 0.57, recall at 0.756 and f1-score at 0.653 which are improvement over the last two classifiers. Also noticeable was the relatively high recall which suggests it predicts benign correctly more of the time.



```
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(train_img_pca,y_train)
y_pred = knn.predict(test_img_pca)

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
print("f1-score:",metrics.f1_score(y_test, y_pred))
hide_toggle()
```

```
Accuracy: 0.6333333333333333
Precision: 0.5740740740740741
Recall: 0.7560975609756098
f1-score: 0.6526315789473683
```

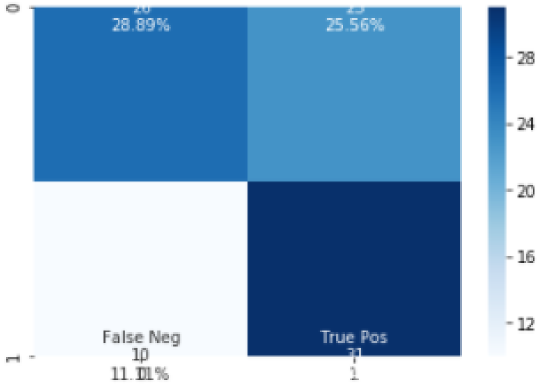
Its cross-validation score of 0.597 was marginally better than the other two which makes it the best classifier in this approach.



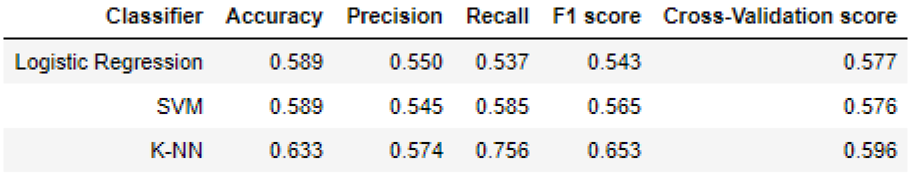
```
validation_score = cross_val_score(knn, train_img_pca , y_train, cv = 10, scoring = 'accuracy').mean()
print('K-NN validation score', validation_score)
hide_toggle()
```

```
K-NN validation score 0.5956926406926406
```

Its confusion matrix is displayed below. It has 31 True positive instances while it has 10 instances of False negatives which explains its high recall score.

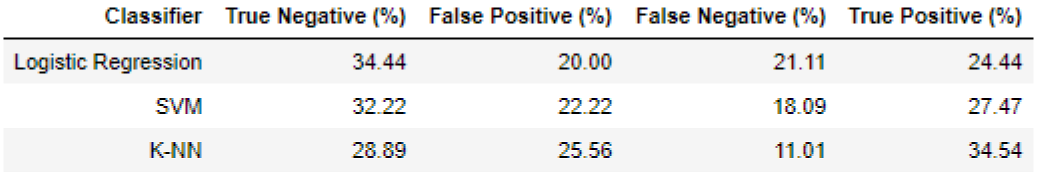


Outlined below is a table summary of the evaluation of classifier on accuracy, precision, recall and cross-validation metrics. K-NN performed best overall while SVM and Logistic Regression were mostly equal in performance.



Classifier	Accuracy	Precision	Recall	F1 score	Cross-Validation score
Logistic Regression	0.589	0.550	0.537	0.543	0.577
SVM	0.589	0.545	0.585	0.565	0.576
K-NN	0.633	0.574	0.756	0.653	0.596

The classifiers were also compared based on their confusion matrices and it was discovered that K-NN was a much better predictor of benign class while Logistic regression was better at predicting the malignant class.



Classifier	True Negative (%)	False Positive (%)	False Negative (%)	True Positive (%)
Logistic Regression	34.44	20.00	21.11	24.44
SVM	32.22	22.22	18.09	27.47
K-NN	28.89	25.56	11.01	34.54

NEW FEATURE ENGINEERING

LITERATURE REVIEW

There has been a number of papers published on classification of skin cancer images using machine learning methods and techniques to process and de-noise dermoscopic images. For machine learning methods, some include;

1. **Deep Neural Networks** **2.PCA** **3.Clustering**

Deep Neural Networks (DNNs) uses novel designed transfer learning based deep neural network like skin_inceptions_v3_nn to help achieve a high prediction accuracy [1]. The melanoma recognition neural network (skin_recnn) is built based on the backbend of Google Inceptions V3 network.

Principal Component Analysis (PCA) The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a strong correlation between observed variables. [2].

Clustering: These methods involve the partitioning of a color (feature) space into homogeneous regions using unsupervised clustering algorithms. [2] Segmentation refers to the partitioning of an image into disjoint regions that are homogeneous with respect to a chosen property such as luminance, color, texture, etc.[2] However, they often lack robustness for low contrast images and may not perform well on complex images that exhibit significant volume of undesirable artifacts.

Image processing and de-noising techniques include:

1. **Color space transformation 2.Contrast enhancement 3.Artifact removal 4.Segmentation**

Color space transformation: The L*u*v color space attempts perceptual uniformity. And then luminance plane is separated. It was previously also used for segmentation of skin lesions. So we have first separated R, G and B planes of RGB image and then converted it to L*u*v color space followed by separation of luminance plane.[4] Due to the computational simplicity and convenience of scalar (single channel) processing, the resulting RGB (red-green-blue) color image is often converted to a scalar image using one of the following methods:[2]

- ***Retaining only the blue channel (lesions are often more prominent in this channel).***
- ***Applying the luminance transformation, i.e. Luminance = 0.299×Red+0.587×Green+0.114×Blue.***

Contrast enhancement: Because of air bubbles and non-uniform lighting conditions during image acquisition step, dermoscopic image can have non-uniform background as well as low contrast. [4] To reduce the effect of above factors, several enhancement techniques can be used in to improve the low contrast of the image and for removing artifacts.[4] It is a technique used to improve the local contrast of an Image instead of improving contrast of whole image. This technique has the advantage of using local information instead of using the entire image; this improves each local area. Enhancement has advantage that lesion will be highlighted against lighter skin so lesion will be identified easily.

Artifact removal: Dermoscopy images often contain artifacts such as such as black frames, ink markings, rulers, air bubbles, as well as intrinsic cutaneous features that can affect border detection such as blood vessels, hairs, and skin lines. These artifacts and extraneous elements complicate the border detection procedure, which results in loss of accuracy as well as an increase in computational time. The most straightforward way to remove these artifacts is to smooth the image using a general purpose filter such as the Gaussian(GF), median(MF), or anisotropic diffusion filters(ADF).[4]

Saliency Based Segmentation: Saliency segmentation computes the most informative region in an image based on human vision perception such that salient and non-salient parts become foreground region (skin lesion) and background region(healthyskin),respectively.[3] It has been alluded that a good saliency segmentation model should satisfy three essential criteria of good segmentation, high resolution, and computational efficiency.[3] Computational efficiency means that saliency-based segmentation methods should rapidly detect salient regions with less complexity.

References

1. Skin cancer reorganization and classification with deep neural network, Hao Chang
2. Automatic skin cancer Images classification, Mahmoud Elgamal
3. Segmentation of Melanoma Skin Lesion Using Perceptual Color Difference Saliency with Morphological Analysis, Oludayo et al
4. Artefact Removal and Contrast Enhancement for Dermoscopic Images Using Image Processing Techniques , Pragati Rajendra Mahajan

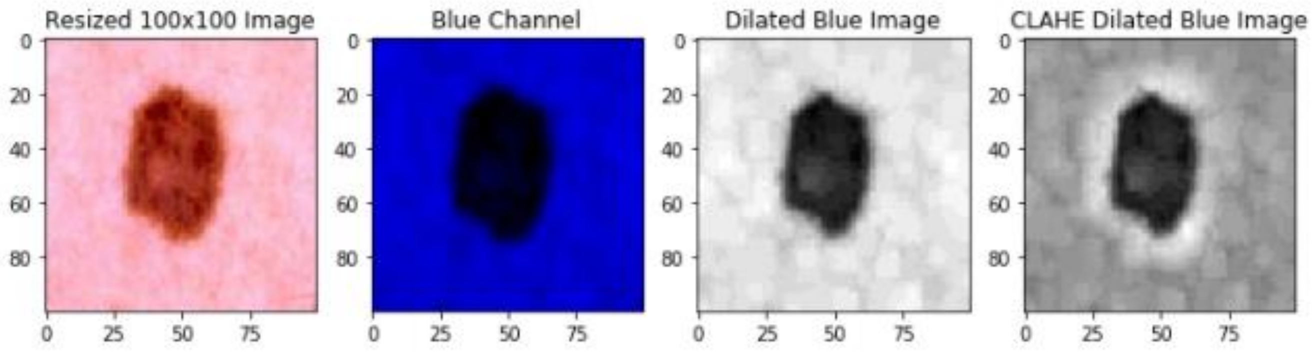
FEATURE ENGINEERING

From literature, it was pertinent that the images would have to be furthered processed and de-noised in order to obtain an improvement in classification performance. Four different steps would have to be followed before final feature extraction and they are discussed below.

All the images were resized to 100 x 100 pixels. This would help reduce the feature dimensions of the images and help improve computational efficiency. This compacts the images and also makes them easier to store.

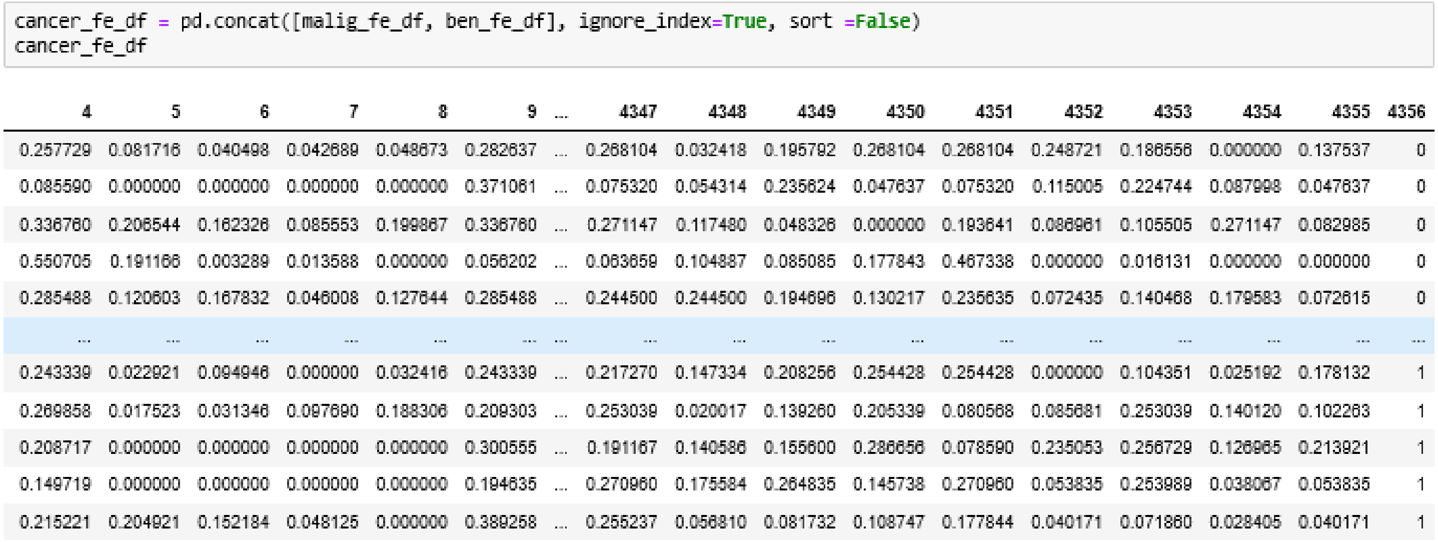
The images were also reduced to only the blue channel as literature explained that lesions are more visible here.

The images were dilated using morphological segmentation in the blue channel. It expands the boundaries of foreground elements and reduces the number of background elements in an image. The output pixel will be the maximum value among all the pixels that fall within the kernel. This will as a result increase the white region in the image. This process also switches the image to grayscale.



After dilating the images, contrast limited adaptive histogram equalization (CLAHE) was applied to improve the contrast of the images locally. This method applies histograms, each corresponding to distinct section of the image and uses them to redistribute lightness values of the image thereby balancing the dilation performed earlier.

The final process to extract out the features from the processed images was by using Histogram of Gradients (HOG). This process extracts the gradient and orientation (magnitude and direction) of localized portions of image edges. The gradients of the images are sensitive to lighting as well and this is reduced by normalizing. The 10000 features image is then reduced to about 4300 features that can be passed into a classifying algorithm. This is performed in on both classes and concatenated into a single dataframe. This dataframe would be split into test and train sets and be passed into the selected classifiers.



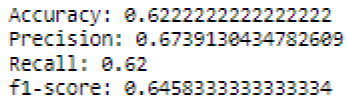
```
cancer_fe_df = pd.concat([malig_fe_df, ben_fe_df], ignore_index=True, sort =False)
cancer_fe_df
```

4	5	6	7	8	9	...	4347	4348	4349	4350	4351	4352	4353	4354	4355	4356
0.257729	0.081716	0.040498	0.042689	0.048673	0.282637	...	0.268104	0.032418	0.195792	0.268104	0.268104	0.248721	0.186556	0.000000	0.137537	0
0.085590	0.000000	0.000000	0.000000	0.000000	0.371061	...	0.075320	0.054314	0.235624	0.047637	0.075320	0.115005	0.224744	0.087998	0.047637	0
0.336760	0.206544	0.162326	0.085553	0.199867	0.336760	...	0.271147	0.117480	0.048326	0.000000	0.193641	0.086961	0.105505	0.271147	0.082985	0
0.550705	0.191166	0.003289	0.013588	0.000000	0.056202	...	0.063659	0.104887	0.085085	0.177843	0.467338	0.000000	0.016131	0.000000	0.000000	0
0.285488	0.120603	0.167832	0.046008	0.127644	0.285488	...	0.244500	0.244500	0.194696	0.130217	0.235635	0.072435	0.140468	0.179583	0.072615	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0.243339	0.022921	0.094946	0.000000	0.032416	0.243339	...	0.217270	0.147334	0.208256	0.254428	0.254428	0.000000	0.104351	0.025192	0.178132	1
0.269858	0.017523	0.031346	0.097690	0.188306	0.209303	...	0.253039	0.020017	0.139260	0.205339	0.080568	0.085681	0.253039	0.140120	0.102263	1
0.208717	0.000000	0.000000	0.000000	0.000000	0.300555	...	0.191167	0.140586	0.155600	0.286656	0.078590	0.235053	0.256729	0.126965	0.213921	1
0.149719	0.000000	0.000000	0.000000	0.000000	0.194635	...	0.270960	0.175584	0.264835	0.145738	0.270960	0.053835	0.253989	0.038067	0.053835	1
0.215221	0.204921	0.152184	0.048125	0.000000	0.389258	...	0.255237	0.056810	0.081732	0.108747	0.177844	0.040171	0.071860	0.028405	0.040171	1

CLASSIFICATION OF MODELS BASED ON NEW FEATURES

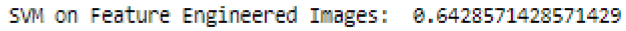
After the feature engineering and processing, the classifiers would be fitted on the train data and would be used to predict the class of the test data. The classsifiers chosen in this instance were Gaussian Naive Bayes and SVM. K-NN was not selected because it would perform poorly given the high number of features. It is expected that they perform better given the feature engineering and processing applied to the image set.

SVM was applied first and evaluated on accuracy, precision, recall and f-1 score, like in the previous method. It scored 0.622 in accuracy, 0.674 in precision, 0.62 in recall and 0.646 in f-1. This was a clear improvement over the previous approach and it's apparent that feature engineering improved the classification. The hyperparameters for the models were also tuned manually and the linear kernel also performs best for SVM.



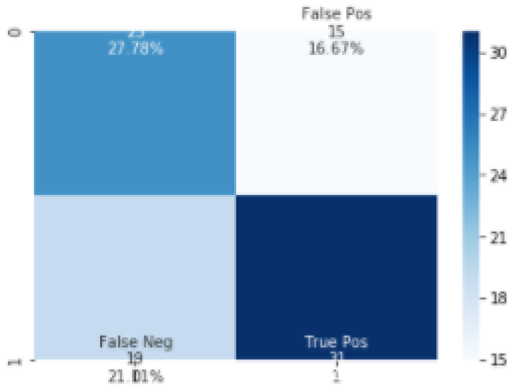
```
Accuracy: 0.6222222222222222
Precision: 0.6739130434782609
Recall: 0.62
f1-score: 0.6458333333333334
```

The cross-validation score for this was 0.6428 which also confirms the improvement based on the feature engineering.

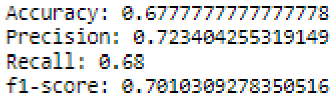


```
SVM on Feature Engineered Images:  0.6428571428571429
```

Its confusion matrix is also shown below showing a true positive on 31 occasions and false positive on 15 occasions. Also 23 true negatives and 15 false negatives.

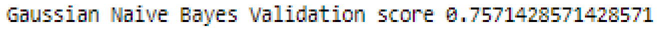


Gaussian Naive bayes classifier which is a parametric classifier and computes its predictions based on priors and assumes feature independence making it highly suitable for this problem. The NB classifier was fitted and tested on the training and test data respectively. It obtained an accuracy of 0.678, precision of 0.732, Recall of 0.680 and f-1 score of 0.701. This is the best result of all the models selected.



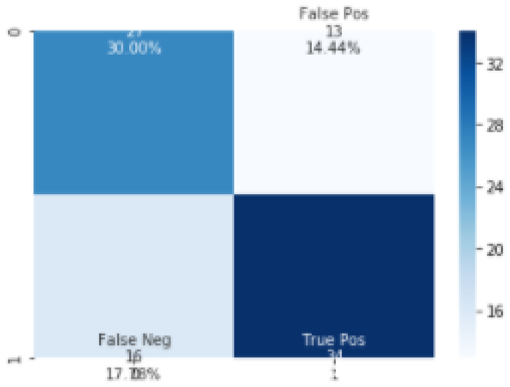
```
Accuracy: 0.6777777777777778
Precision: 0.723404255319149
Recall: 0.68
f1-score: 0.7010309278350516
```

It performed best also in its validation score as well with a score of 0.757. This is extremely good relative to the CV scores of the other classifiers.

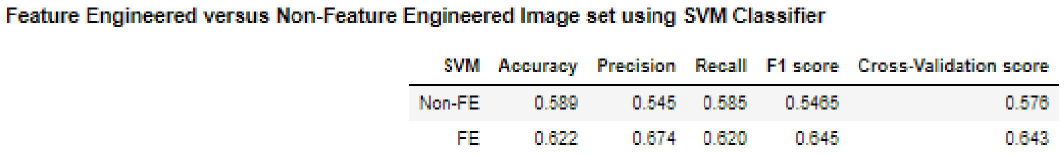


```
Gaussian Naive Bayes Validation score 0.7571428571428571
```

The confusion matrix is displayed below and it performed well in both the true positive and true negative quadrants.

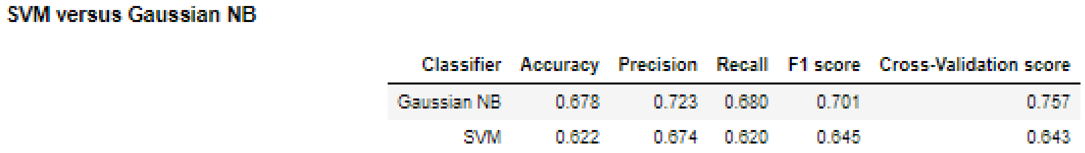


Below is a summary of the classification performance of the classifiers.



Feature Engineered versus Non-Feature Engineered Image set using SVM Classifier

SVM	Accuracy	Precision	Recall	F1 score	Cross-Validation score
Non-FE	0.589	0.545	0.585	0.5465	0.576
FE	0.622	0.674	0.620	0.645	0.643



SVM versus Gaussian NB

Classifier	Accuracy	Precision	Recall	F1 score	Cross-Validation score
Gaussian NB	0.678	0.723	0.680	0.701	0.757
SVM	0.622	0.674	0.620	0.645	0.643

CONCLUSION

Even with a small dataset of images (300) and without neural networks, performing novel image processing and de-noising methods on the image dataset and using supervised machine learning classifiers provided a cross-validated accuracy of 0.757. This shows true potential in disease detection and prediction.

The results also show that the most important characteristics for identifying malignant moles are the intensity and shape of the lesions. There would certainly be an increase in accuracy if a larger dataset was used.

Further research could also be done on how to better de-noise and process images and this would go a long way in improving AI in all fields.