Skin Cancer Diagnostics

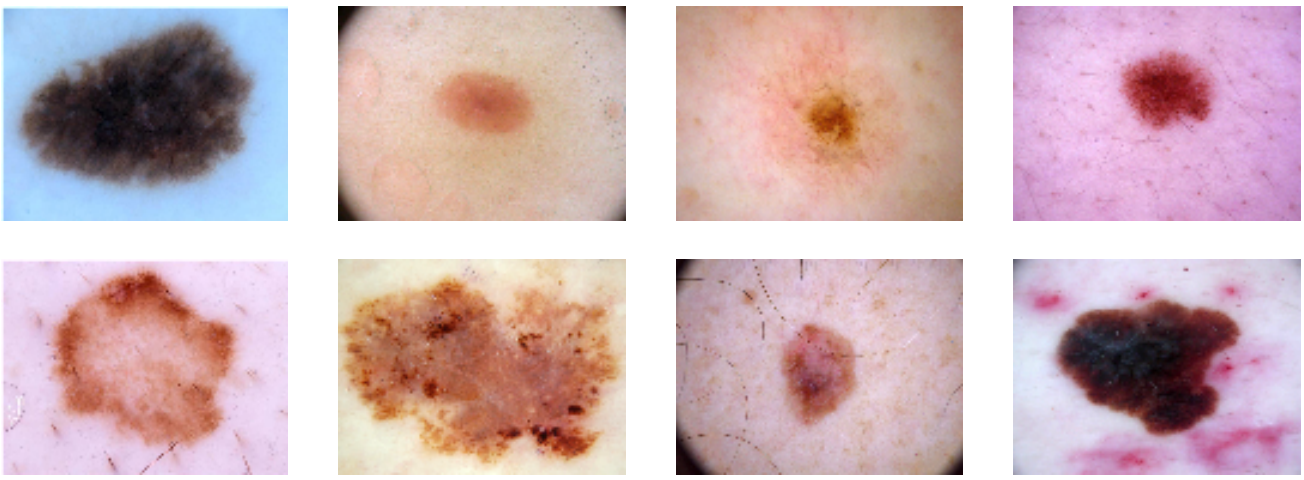
Hao Tang

Contents

I Project Description and Summary

Skin cancer, as the abnormal growth of skin cell is one of the most common cancer in the united states[1] [2] [3]. There are three major types of skin cancer: basal cell carcinoma, melanoma and squamous cell carcinoma. The curing effect to them is based on whether there's detection in the early stage. Regular skin self-exams is very important to detect the signs of skin cancer and it's recommanded by American Academy of Dermatology. However, it is challenging to discover the sign of skin cencer in the early stage without attention and specific training [4]. This project is to explore the detection methods according to skin cancer lesion area with different machine learning models. The project has two sections. The first part is to fit the classification on the skin cancer image pixel numeric value, and analyze the prediction result in different models. The second part is to extract visible features from the images after effective image processing. These features are closely related to comman skin cancer examination methods. The fitting classifiers based on our new features will be more applicable and interpretable to skin cancer diagnosis.

Figure 1: Lesion Image Sample



II Data Processing

In the dataset of this project, we have 300 dermascopy images for different types of skin cancers leision. Half of the images are labelled as confirmed benign and the other half is labelled as confirmed malignant. We can take a look at the image samples in Figure 1. Some of the images are very misleading especial to the people without specific training. The top row is benigh samples and the bottom row is malignant samples. With no indentification/feature to specific type of skin cancer, it seems chanlleging to tell why they are lebelled as it. In following sections, this project will try to analyze the samples with classification model (0 is the label to benign and 1 is label to malignant for all models) by the pixel value and some interesting features.

[1] Mayo Clinic.Skin cancer. https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605
[2] American Cancer Society. Skin Cancer. https://www.cancer.org/cancer/skin-cancer.html
[3] American Academy of Dermatology. Skin Cancer. https://www.aad.org/media/stats-skin-cancer
[4] June K. Robinson. et. al. Skills Training to Learn Discrimination of ABCDE Criteria by Those at Risk of Developing Melanoma. Arch Dermatol Vol 142, Apr 2006

Table 1: Importance Percentage of PCA

	PC1	PC2	PC3	PC4	PC5	PC296	PC297	PC298	PC299	PC300
Percentage	0.27688	0.21348	0.07171	0.05153	0.03646	4e-05	4e-05	4e-05	4e-05	0

Classification to the raw images is very challenging task. Our sample images are captured in different magnification, exposure and distance. Some of them are captured with high-resolution dermoscope whereas others are from normal dermoscope. It seems that we are supposed to have customized image processing according to different image conditions. And this will lead to a very inefficient procedure. Considering the balance of efficiency and classification performance, I decided to resize all the images to the same pixel size and keep original color channels, shadow and contrast to the images. That is, there is only resolution change to the images in pixel-based model pre-processing. Most of the challeging job in classification will be left to our models. The major reason to resizing is becasue some of the large images (about 17Mbs) will dramatically decrease our model's efficiency. Processed images will be with 120 X 90 pixels which is proportional compressed version to the most comman resolution (600 x 450) in the images dataset.

Now, each processed image has 32400 (120W X 90H X 3C) elements, which means 32400 variables/predictors in classification fitting, still pretty large to the following model processing. Quite a lot of pixels are baraly informative, e.g. the pixel close to the image's corner and edge. On ther other hand, the pixels in the center of the lesion area are crucial since they may represent importance features. In order to focus on the most important feature, I use PCA to transform predictors from orginal pixel space to PCA space. After PCA transforming, we will have 300 X 300 matrix for 300 samples and 300 principle components as the new predictors. Please notice when $n << p$, we need to pay attention to the importance of the variable after transform, otherwise the result won't be stable[5]. Table 1 shows the importance to the first 5 and last 5 principal components. We can see percentage to the last 5 principal has decreased dramatically thus our PCA transformation should be stable and reasonable. Even though we can further decrease the predictor number according to the importance value, all the 300 predictors will be kept in the model fitting section so the fitting model will be constructed on top of all the principle components. Our classifiers should be able to handle the whole information we got after data-processing.

Another popular image processing method is to keep the green channel value and resize the picture to a relatively lower resolution[6]. I didn't apply this method in the report but the general model performance with this processing is slightly worse than the method applied here and next section.

In the feature engineering section, I will apply more advanced procedure to the dataset based on the diagnosis criteria and this isn't the focus to pixel value classification section[7].

III Classification Models Based on Pixels

K-Nearest Neighbors

A model may be the most important part of machine learning. We build a model with training data and use the model to describe the data status and predict the output for future input data. After data processing, we are ready to model construction.

First, let's try the most straightforward model, k-nearest neighbors (KNN) classifier. This classificatin model search for the k training observations that's closest to the test observation then apply major vote based on the labels of these k observations. The resulting label from the vote will be the label to the test observation. The closest distance usually is based on Euclidean distance. KNN made very mild assumption in its structure so it can start to work with or without model constuction/training! Because of its simple algorithm, KNN is very popular and efficient to implement. In this section, I test the training accuracy and testing accuracy on different degree of freedom (N/k, given N is the total number of training observations) by 3-fold cross validation so our result will be more stable.

Figure 2 shows trend of training accuracy and testing accuracy as degree of freedom increase. The plots are jagged due to our small training and test set size(3-folds cross validation on 300 images) and the image classification difficulty. However, we can still see the testing accuracy increases at first as degree of freedom increases, then it decreases after some optimal values. As for the training accuracy, the trend is going up as degree of freedom increases. Our observation makes sense according to the bias-variance tradeoff. A large k means stable fit and high bias and low variance, and the overall accuracy is relatively low, which is shown on the left of Figure 2a. In contrast, a small k means flexible fit, as well as low bias and high variance, thus the overall accuracy is all low. There exists a gap between large k (Or small degree of freedom) and small k (Or large degree of freedom) where we can observe the optimal test accuracy. The best accuracy to the KNN model is 0.69 when `k = 4`.

[5]Jeongyoun Ahn. et al. The high-dimension, low-sample-size geometric representation holds under mild conditions. Biometrika (2007), 94, 3, pp. 760–766

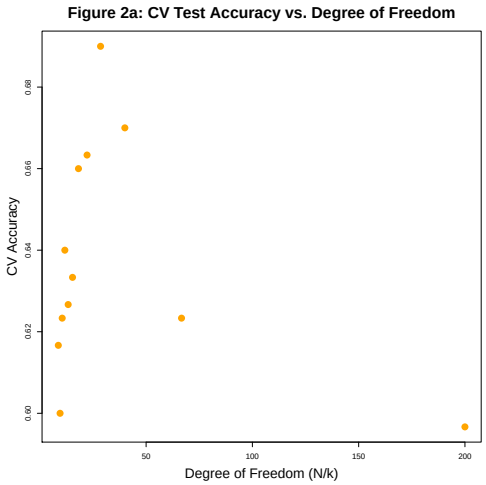
[6]J.L.Reimers. et al. Green Channel vs. Color Retinal Images for Grading Diabetic Retinopathy in DCCT/EDIC.Investigative Ophthalmology & Visual Science April 2010, Vol.51, 2285.

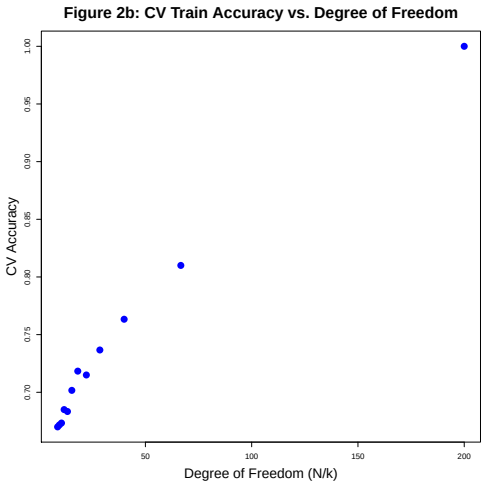
[7]Andre Esteva. et. al. Dermatologist-level classification of skin cancer with deep neural networks. Nature Vol 542 February 2, 2017.

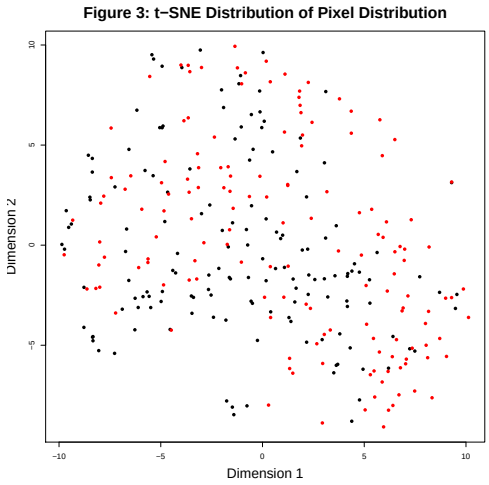
Table 2: Final KNN Model

	Test Accuracy	K
KNN Classifier	0.69	7

KNN is actually based on the Euclidean distance between the datapoints. We can take a look at the data distribution with t-SNE (t-distributed stochasitc neighbor embedding) on the original pixel matrix [8]. Figure 3 shows that we are fitting models on a challenging dataset apprently.







Support Vector Machine

Since our images are with binary labels, support vector machine (SVM) classifier seems to be good fit. SVM emphasize on construct separating line or hyperplane between classes. The binary classes may be difficult to seperated in low dimensional space but it can always be separated in high dimention. SVM can deal with this type of classfication by projecting the observations' similarity to space with huge or infinite dimension. The projection is implemented by the kernel function. A kernel is mapping that projects the low dimensional space data's relationship to high dimentional space. Without knowing the direct projection of the observation itself, the kernel help us deal with the similarity in high dimensional space and separate the observations.

Similar to KNN classifier, cross validation shall be applied to estimate accuracy. Incredibly, package `e1071` comes with a `tune()` function which by default performs ten-fold cross-validation on the input dataset with tuning parameters input. That being said, by inputing combinations of tuning parameters (or a parameters' grid), I can achieve the corresponding CV error to corresponding models. So instead of separating train set and test set, I use the whole dataset in tune function to estimate the optimal model and let the tune() function to take care of the cross validation.

As for the kernel option, we can choose polynomial kernal, linear kernel or radial kernel. Since I don't know data distribution in the pixel space, I use radial kernel in case the data distribution comes with curvy boundary. For radial kernel function, we will tune two parameters to search optimal model: the cost C and γ.

Table 3 presents the comparison between different combinations of cost and γ. According to the accuracy, we have the best setting : gamma= 0.005, cost = 1.2 (Table 4). It seems like the best model is with small cost and very small γ. Large value of cost will eliminate positive slack parameter ξ and make the boundary wiggly. A small cost means the model can perform well with normal tolerance to misclassified data around the dicision boundary. That is, the boundary shall be smooth. A small γ means the decision boundary doesn't needs to be curvy as well, which means a linear boundary may also work well.

In order to compare the radial kernel and linear kernel performance in our dataset, I separated the dataset into 200-sample train set and 100-sample test set and fit the models with two kernels accordingly. For the radial kernel model, I used the parameters achieved from the radial kernel cross-validation. For the linear kernel model, I used the same cost value (1.2) and kept default values to other parameters (We don't have tune γ in linear kernel model). Figure 3 shows the ROC curve to the radial kernel classifier and linear kernel classifier. Their curves have pretty close shape. According to their AUC (area under the curve), these models have similar classification ability in different thresholds.

[8]Laurens Van der Maaten et. al. Visualizing Data using t-SNE. Journal of Machine Learning Research. 9 (2008) 2579-2605

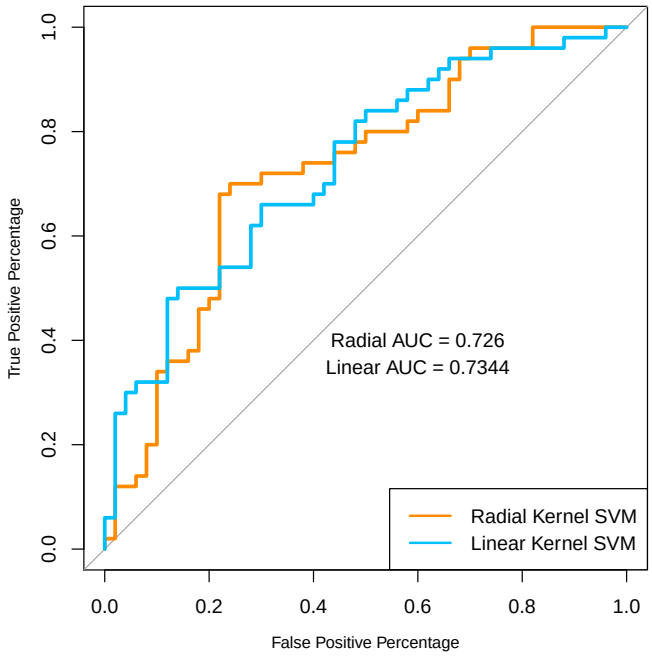
Table 3: Accuracy Comparison on 10-fold Cross Validation

	point1	point2	point3	point4	point5	point6	point7	point8	point9
gamma	0.0040000	0.0050000	0.006000	0.004000	0.0050000	0.006000	0.0040000	0.0050000	0.0060000
cost	1.0000000	1.0000000	1.000000	1.200000	1.2000000	1.200000	1.4000000	1.4000000	1.4000000
accuracy	0.7487308	0.7492364	0.749227	0.749159	0.7492367	0.749227	0.7491591	0.7492367	0.7492271

Table 4: Final SVM model

Kernel	cost	gamma	CV Accuracy
radial	1.2	0.005	0.7492367

Figure 4: ROC to SVM model



Random Forest

We can try a very different model in this part. Let's investigate the performance on randomforest based on new observations from PCA transformation. As for the high dimension of the orginal dataset (120W X 90H X 3C), random forest algorithm fitting and prediction are very expensive and they probably cannot fit in most computer's memory. After PCA, our data has been transferred to a 300-dimensional data so the random forest model is accessible.

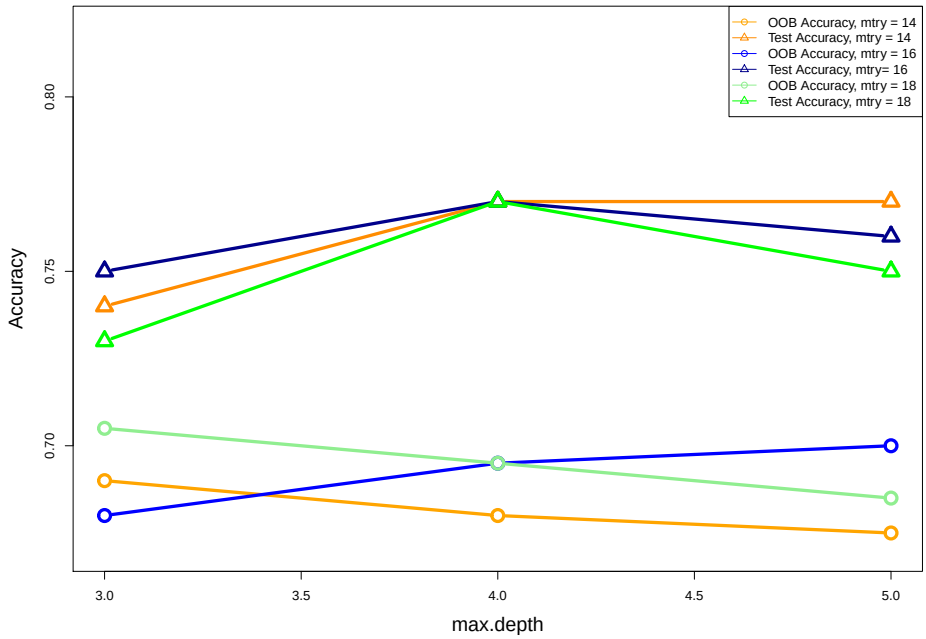
Random Forest is ensemble model constructed on top of multiple dicision trees. It applies bagging or bootstrap aggregating idea and construct the decision trees (forest) from bagging. When we make prediction on a datapoint/observation, the regression result will be generated by the trees' result averaging and the classification result will be generated by the trees' major vote. The most important advantage of random forest is correlation reduction in the model fitting process. Which exchange lower variance by adding a bit bias. Particularlly, in each node when we want to grow the tree, only some of the variables (**mtry** as below analysis) will be selected as candidates. Then we chooce the best split value in these candidates as the criteria for the next level's tree. This process will be processed recursively in each level of the tree until we hit the maximum depth (**max.depth** as below anlaysis) or minimum node size of assigned observations. For another, we can always increase the depth of each tree to decrease the bias which will improve the final prediction accuracy just like ordinary decision tree.

In this section, I chooce different values of **mtry** and **max.depth** to verify the corresponding model's test prediction accuracy (With 100 image as test set). One of the important feature of random forest model is its out-of-bag (OOB) error estimate. This OOB error estimate is pretty close to N-fold cross-validaion performed during model construction. So it will be our another reference.

Table 5: Final Model of Random Forest

mtry	max.depth	OOB Accuray	Test Prediction Accuracy	Tree Number
16	4	0.695	0.77	500

Figure 5: Random Forest Accuracy with Tuning Paramters



We can see the accuracy displayed in Figure 5. The OOB accuracy is generally lower than the test sample prediction accuracy. This make sense because the training set for the OOB accuracy is smaller than the training set for the test sample. We can see there is little change to the accuracy when the **mtry** is changed since we fit the random forest on the data after PCA, which will decrease the correlation between the principal components. As the **max.depth** increase from 1 to 3, both OOB accuracy and test accuracy will increase fast. This means the model performance will improve when our model complexity increase. After 3, both accuray do not change any more but stay around certain value (see Figure 4). This means the depth of tree has little influence to the test accuracy after we grow the trees to certain depth. However, the training accuracy will continue to increase unitl it hits 1, which means the model is overfitted even though it won't affect the test accuracy dramatically. Considering the model efficiency and prediction accuracy, I choose **max.depth** = 4 and **mtry** = 16 as the final model's parameters (Table 5).

Based on the fitting result, we can argue that the random forest classification can be compatible with the 300-predictors dataset even though it's inefficient to the original dataset.

IV Literature review

We can definitely futher improve pixel-based model in previous section to achieve better performance. However, the project also emphasize on exploring the computer-aid diagnosis methods that's more intepretable to our customers, dermatologists. The pixel-based model we fitted in last section is basically values analysis to thousands of elements from sample images. Indeed, with previous models and the direct pixel values, we don't need to care how the image looks like visually. The image is the input to the black box and customers use the model to generate output label for diagnosis. That being said, pixel value classification process seems like a powerful machine learning application more than a handy analytic tool for skin cancer diagnosis. But machine learning is also powerful of mimicing human-being analysis procedure. The learning process and prediction preocess can be constructed by domain expertise in skin cancer detection effectively. The second part of this project will focus on building classification model on dermatologic diagnosis intimately.

Let's refer to the popular methods used in skin cancer examination as the beginning to this part. More than thirty years ago, dermatologists have recommanded the well-known ABCD rule for skin cancer detection and self-examination[9] [10] [11]. Table 6 displays the detailed ABCD rule from Skin Cancer Foundation Website [12]. Generally, we are supposed to exam

[9]Robert J. Friedman. et. al. Early Detection of Malignant Melanoma:The Role of Physician Examination and Self-Examination of the Skin. CA-A Cancer Journal for Clinician Vol.35,No.3 May/Jun 1985.

[10]Robinson JK Rigel et. al. What promotes skin self-examination? J Am Acad Dermatol. 1998;38:752-757.

[11]Thomas L. et. al. Semiological value of ABCDE criteria in the diagnosis of cutaneous pigmented lesions. Dermatology 1998;197:11-17.

[12]https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/

Table 6: The ABCD Rule to Skin Cancer Diagnosis

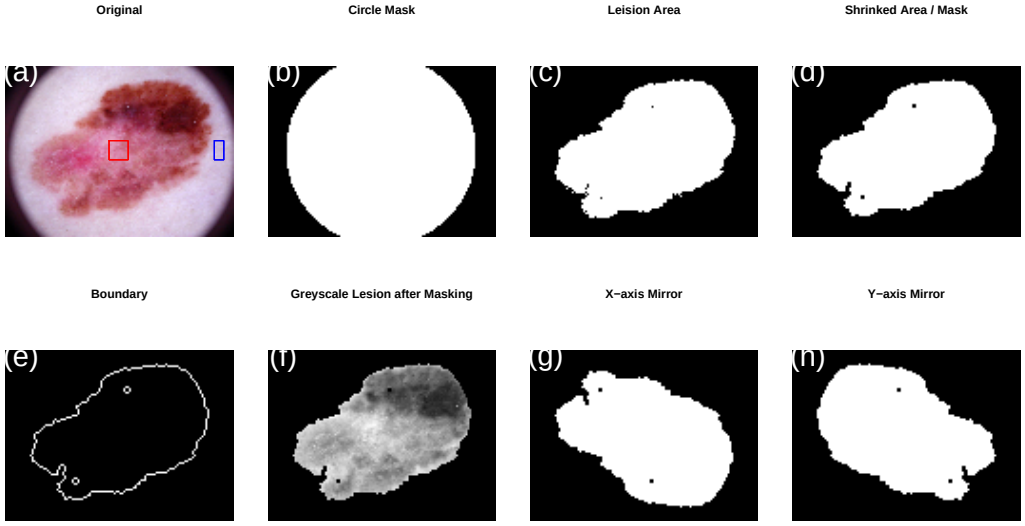
Rule	Description
Asymmetry	Most melanomas are asymmetrical. If you draw a line through the middle of the lesion, the two halves don't match, so it looks different from a round to oval and symmetrical common mole.
Border	Melanoma borders tend to be uneven and may have scalloped or notched edges, while common moles tend to have smoother, more even borders.
Color	Multiple colors are a warning sign. While benign moles are usually a single shade of brown, a melanoma may have different shades of brown, tan or black. As it grows, the colors red, white or blue may also appear.
Diameter/Dark	While it's ideal to detect a melanoma when it is small, it's a warning sign if a lesion is the size of a pencil eraser (about 6 mm, or ¼ inch in diameter) or larger. Some experts say it is also important to look for any lesion, no matter what size, that is darker than others. Rare, amelanotic melanomas are colorless.

the suspected area in our skin periodically whether it's symmetric with a line crossing the center, whether its border is smooth or jagged, whether its color is even, whether its small and apparently darker to its neighbor mole/skin. This rule is primitively generated by Melanoma early detection which is critical to this most fatal skin cancer but it has become the rule of thumb to all epidermal and melanocytic malignant detection. For decades, doctors and cancer research organization have made great effort to apply and distribute this knowledge to everyone[13]. As the prerequisite of new classification models, later analysis will closely follow these rules to generate new features from the image dataset.

V Feature engineering

Before we start this section, let's come back to our sample images. Our images includes benign and malignant mole, like nevi(benign), seborrhoeic keratoses, squamous cell carcinomas, Basal cell carcinomas and melanomas. Even though they probably have its own chracteristics to the lesion area, it's difficult for people without training to identify the feature, sometimes it's challenging for dermatologists to confirm their diagnosis without biopsy exam. This is mostly because our raw eyes and recognition system may miss some important information from the images.

Figure 6: Feature Extraction Process



In this section, an image processing package `imager` in R is applied to image loading, feature extraction. I also use dedicated algorithm simultanesouly to make this procedure more efficient. Here is detail. The raw images will be loaded by the imager loading/resizing function. Some of them may come with the dark corner (Figure 6 (a)) that's generated by the round circular lens designed for smaller sensor. I first apply a mask (Figure 6 (b)) to the picture to exclude these section from our Area of Interest (AOI). Then I choose two areas as the lesion area (red square in the middle of Figure 6 (a)) and background skin area (blue rectangle in Figure 6 (a)). Their pixels are labelled by assigned class (lesion or skin). A 1-KNN algorithm is then applied to the whole image area with previous two areas as the training set. With this algorithm, we will effectively identify the lesion area from the image background(Figure 6 (c)). In order to make this work easier, all

[13]Pauline F. Hanrahan. et. al. The effect of an educational brochure on knowledge and early detection of melanoma. Australian Journal of Public Health Volume 19, Issue 3, June 1995, Pages 270-274

the images will be processed to same resolution (110W x 90H x 3C) and the lesion area will be consistently centered in the image prior to the KNN method. Because 1-KNN method is very sensitive, the boundary is sharp and probably too curly. So I will shrink the detected area with a small value(Figure 6 (d)). Then the image will go through normal boundary detection function to generate the boundary (Figure 6 (e)). Meanwhile, the detected lesion area is applied to the greyscale picture as a mask (Figure 6 (f)). Figure 7 shows some samples with their lesion identification. After this effective image processing, we are ready to generate our diagnosis features:

Asymmetrics:

Figure 6 (g) and (h) show the lesion area mirro about the X-axis and Y-axis. For consistency, the asymmetrics will be sum of X-axis mirror images area difference and Y-axis mirror images area difference for all images.

Border:

We can conviniently extract the boundary length and the lesion area from the output image in Figure 6. In fact, we care about the relationship between border and the lesion area. That is, how curly the border is to the lesion area. The most effective metric to this is compactness[14]

$$Compactness = \frac{(Border)^2}{4 * \pi * Area}$$

This is the feature that described whether the lesion border is uneven or scalloped (Table 6). In Figure 7, the compactness of these are 2.1326953, 12.7542094, 3.4811455 and 1.4190021.

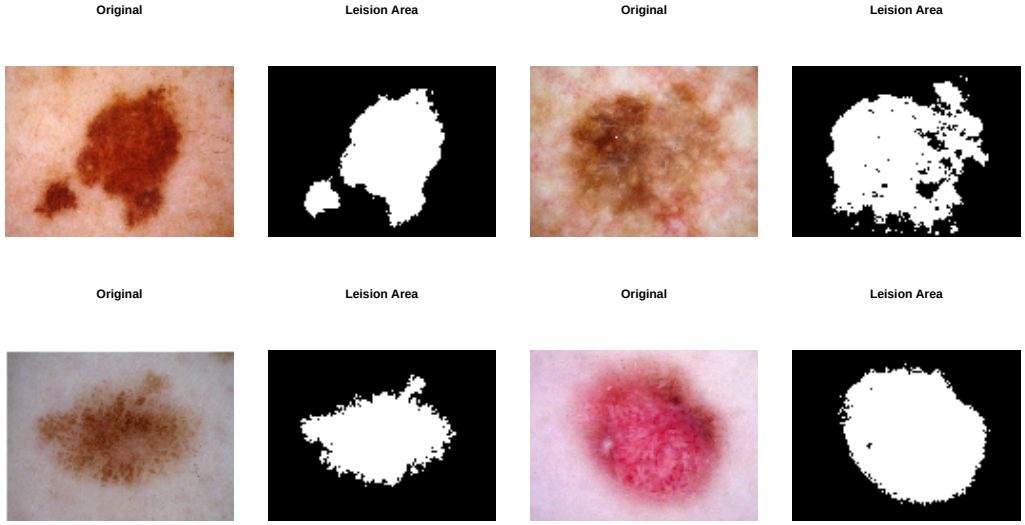
Color:

Greyscale difference can be calculated in the output images with the lesion mask (Figure 6 (f)). Here I use the greyscale difference between the top 10 values average and bottom 10 values average to prevent noise influence. For another, greyscale can alleviate impact from luminence fluctuation since the three color channel is merged to get the greyscale.

Diameter/Dark:

Lesion area will be our new feature as well. However, it shall be noticed that the area size will be closely related to the image capture distance and magnification which may be quite different between samples.

Figure 7: Lesion Detection Samples



VI Classification models based on new features

With the new generated features, we are ready to construct new classification model that's highly related to the dermatologistic diagnosis. Let's first glimpse our new features before we fit the models. Table 7 presents the covariance to the ABCD features generated above. Interestingly, The **lesion.area** seems to positively related to **asymmetrics**,

[14]Lawrence H. Staib et. al. Boundary fitting with parametrically deformable models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 11, Nov 199.

Table 7: Covariance of Features

	asymmetrics	compactness	color.difference	lesion.area
asymmetrics	1.0000000	0.4829406	0.2641701	0.6413923
compactness	0.4829406	1.0000000	0.0246999	0.0957222
color.difference	0.2641701	0.0246999	1.0000000	0.4190995
lesion.area	0.6413923	0.0957222	0.4190995	1.0000000

Table 8: Confusion Table to Logistic Regression and LDA

	Prediction	True 0	Ture 1		Prediction	True 0	Ture 1
	LR.pred.0	40	18		LDA.pred.0	40	18
	LR.pred.1	10	32		LDA.pred.1	10	32

which means in our dataset, the large lesion is more likely to be assymmetric. Also, compactness is positively related to assymmetric as well. This makes sense because the more complicated/uneven the boundary is, the less likely to see the symmetric lesion.

Logistic Regression

As a straightforward model, logistic regression is a good fit to our binary class dataset. Logistic regression is generally a linear classification method that applied the similar idea from linear regression. The response is transformed probablity that's projected to another space by ligit link function. The predictors coefficient is fitted based on the link function value and linear combination of predictors. We can apply similar estimation to the logistic regression model directly. So, similar to pixel-based classification model, the dataset will be firstly separated into train set (200 samples) and test set(100 samples) before model fitting. And 0 is assigned to benign and 1 is assigned to malignant. Below is the summary to the logistic regression model coefficients:

```
coef(summary(glm.fit))
```

```
##                      Estimate   Std. Error     z value    Pr(>|z|)
## (Intercept)      -1.281915e+00 0.5991081138 -2.13970595 0.032378538
## asymmetrics       1.013402e-05 0.0001755824  0.05771663 0.953974346
## compactness       1.031409e-01 0.0371029298  2.77985991 0.005438235
## color.difference  1.280933e+00 1.5306214091  0.83687146 0.402664824
## lesion.area       1.530509e-04 0.0001793076  0.85356623 0.393345377
```

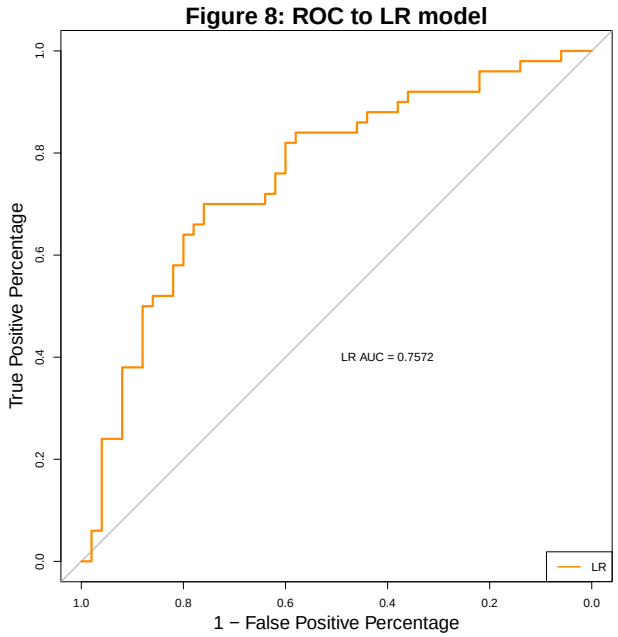


Figure 8: ROC to LR model

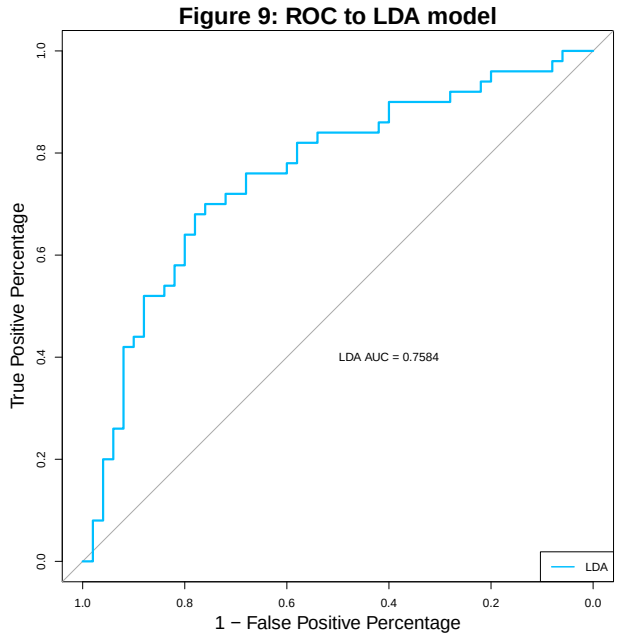


Figure 9: ROC to LDA model

The model coefficient shows that only the **compactness** feature is significant in the logistic regression model. This is related to the consistency issue in our sample images. The color and area size can be quite influenced by picture capture

Table 9: CP Table of Classification Tree

CP	nsplit	rel error	xerror	xstd
0.3854167	0	1.0000000	1.1041667	0.0735243
0.0833333	1	0.6145833	0.7187500	0.0700284
0.0312500	2	0.5312500	0.6979167	0.0695308
0.0104167	3	0.5000000	0.6770833	0.0689981
0.0052083	5	0.4791667	0.7604167	0.0709214
0.0000000	9	0.4583333	0.7916667	0.0715042

process but compactness is less likely to be affected since it's the description to the complexity of lesion boundary. The asymmetrics is also not significant in the logistic regression model. This is because we can only get the asymmetric level by x axis and y axis mirror images.

As for the similarity of logistic regreesion and linear discriminant analysis(LDA), I will also present the result from LDA for comparison. Both models are fitted on the same train set and test set. After model fitting, we have the classification accuracy to the logistic regression and LDA:

– Logistic Regression : `0.72`

– LDA : `0.72`

They are the same. As for the confusion table of the classification (Table 8), the result are the same. Figure 8 and Figure 9 are the ROC to the logistic regression model and LDA model. They are also very close to each other. This proved our knowledge to these models. The relationship between response and coefficient are the same. The only difference between these procedure are the estimation approach.

Decision Tree

As for the interpretability, decision tree will be on top of regression and classification model. Tree algorithm is based on the split rule in each node. From the root to the level on top of leaf node, we recursively search a split variable and cutting value to split the observation and assign observations into corresponding nodes. Finally we will reach a point where we don't want to split the observations any more and they'll be kept in specific leaf node. This is tree-growing procedure. The full tree will be pruned to prevent overfitting and decrease the variance. Generally, decision tree is straightforward for application in diagnosis and easy to learn. It's also very powerful tool to present the skin cancer detection process to the patients. Here I will investigate the performance to the tree model. The following single tree model is fitted by the rpart package. Figure 10 is the full-grown tree and the sample percentage in each split. Pruning a tree is as important as, if not more than, the tree-growing procedure. In this section, cp value and its corresponding cross-validation error(xerror) is the pruning reference. The minimal xerror is located from the cp value table, the tree will be pruned to the level just on top of the level with minimal xerror. By doing so, we exchange the lower variance by a little more bias and more stable model. Figure 11 (Here 0 means benign and 1 means malignant) shows the pruned tree model with the observation class percentage in nodes of each level.

Figure 10: Whole Tree with NO Pruning

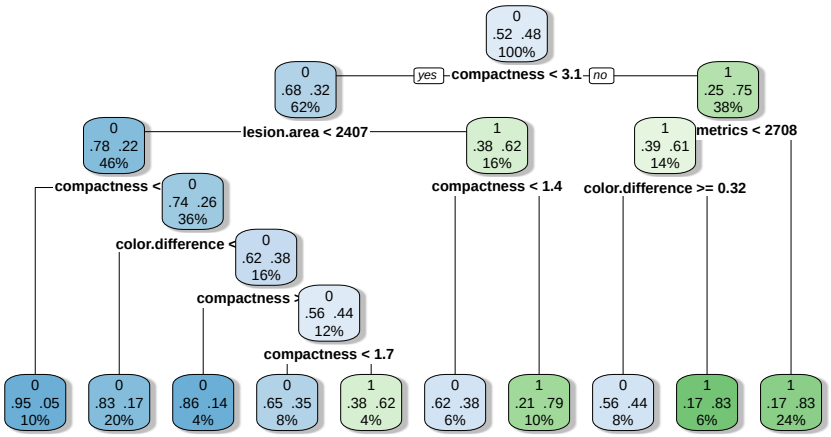
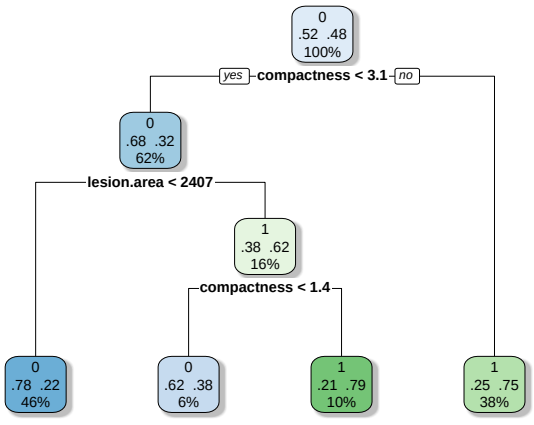


Figure 11: Pruned Tree with minimal Xerror



The test accuracy is 0.66. It's relatively lower comparing to other classifier we generate in previous sections. However, we are able to analyze our dataset by a straightforward tree model by the explicite spliting rule. From the pruned tree

model in Figure 11, we can see the compactness is also the important parameters in our split rule, which matches our findings in logistic model.

Summary

From the prediction accuracy, we can conclude that the extracted features are effective to be the predictors in our models. With very basic procedure and little parameters tuning, we have made remarkable progress that's comparable to the specialists' work twenty years ago. The **conpactness** feature is more important than other features in model prediction. The logistic regression model and decision tree model fitted in this section is basic application to the feature we generated. More sophisticated learning process is available immediately although they may not be as intepretable as the models we presented here.

VII Conclusion

This project presents two types of classification methods on the skin cancer sample images. The first one is to contruct classification model based on the pixel value, which is closely related to learning model mechanism, and model performance in theory. The second one is based on more advanced image processing or feature engineering that's on top of the sample images, together with specialists' knowledge and domain expertise. Multiple classification models are applied in these sections. Some common statistical learning analysis are utilized to evaluate these model's prediction accuracy and related properties. Further exploration to above content should be ready for next action.