Classification of diabetic retinopathy

Tarang Mahapatra, Gordon Ng

Abstract

In this project, we aim to classify images from the APTOS 2019 Blindness Detection dataset from Kaggle. Our dataset has 5 ordinal classes corresponding to the progression of diabetic retinopathy (DR) - no DR, mild, moderate, severe and proliferative. Our dataset is large (9.52 GB) and consists of 3662 labelled images with each image ~2000x2000 pixels. In our project, we tried 3 methods - logistic regression, pixel-wise random forest and convolutional neural networks. In terms of accuracy, the CNN performed the best followed by the random forest and the logistic regression performed considerably worse. This project showed that while CNNs perform better on tasks with image datasets as compared to traditional statistical methods, they have drawbacks as they are less interpretable and consume much more computational resources.

Introduction

Diabetic retinopathy (DR) is a complication in the eyes caused by damage to the blood vessels in the retina [1]. There are various degrees of severity to this condition resulting in mild to severe problems with vision. Diagnosis of this condition is performed through a visual inspection of the eye and the retina through an eye exam and optical coherence tomography (OCT). The goal of this project is to classify the severity of diabetic retinopathy among patients from a given dataset of OCT scans. The OCT scans are cross sectional images of the retina, which encompasses features of the blood vessels and retinal tissue.

The dataset is obtained from Kaggle and contains 3662 labelled images. Only the class labels of each image have been provided; there are no explanatory variables. The classes are ordinal and are labelled from a scale of 0 - 4 on the severity of the condition. Class 0 represents a healthy retina, with no diabetic retinopathy, class 1 represents a mild case, class 2 is considered moderate, class 3 is severe, and class 4 is proliferative. An example of each image has been shown below in FIG.1.

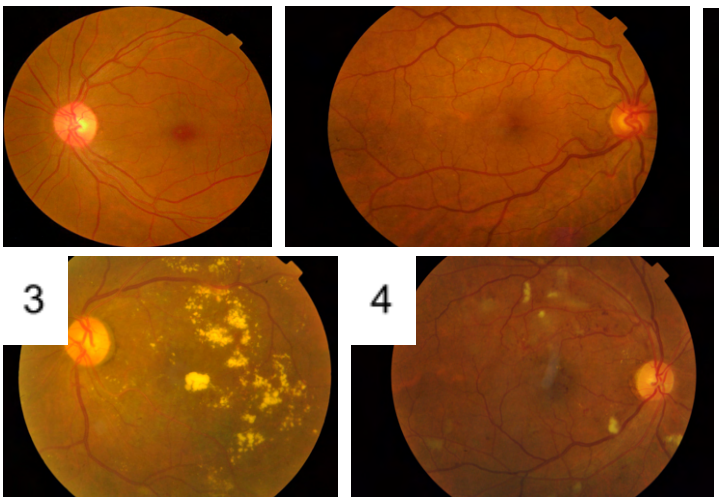

FIG.1: Retina images from the dataset, with their class label indicated on the top left.

Dataset Images

The images provided represent an unequal class distribution with 49% of the images representing class 0 (where diabetic retinopathy is non-existent) (refer to FIG.2). Among these images, there are also a variety of different lighting conditions as can be seen in FIG.3, we have 3 images of the same class with vastly different colours. To investigate further on this issue, an analysis on the color distributions has been performed. Color images can be represented by their HSV (hue, saturation, and value) components for each pixel. In FIG. 4 we have boxplots of the mean HSV values for each image. It is apparent that there is a noticeable discrepancy in the distribution of hue for class 0. For the saturation and value components, it appears that they are distributed roughly equally among the 5 classes.

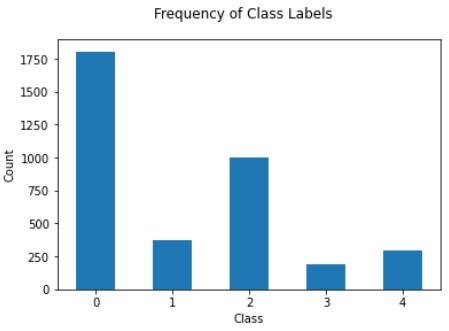

FIG. 2 Class distributions of the dataset

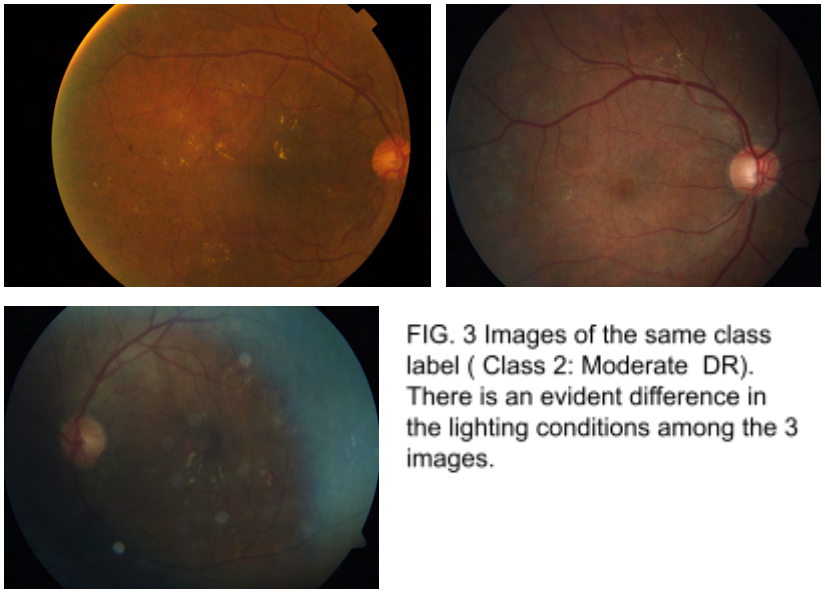

FIG. 3 Images of the same class label (Class 2: Moderate DR). There is an evident difference in the lighting conditions among the 3 images.

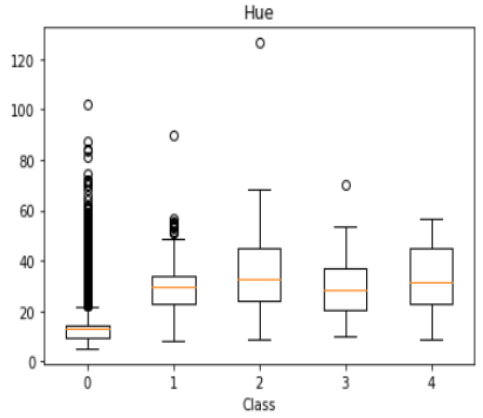

FIG. 4 Boxplot of the saturation of the images. There is a noticeably different distribution for class 0.

Variables, Transformations and Preprocessing

To improve the accuracy and efficiency of our models, we perform a series of transformations and preprocessing to the images. As noted earlier, the dataset contains a wide range of images with various imaging conditions. Some images are off center, out of focus, contain lighting distortions (such as glare), and vary in colour based on the lighting. We start by cropping the edges of each picture such that the retina image is centered and to remove as much of the background as possible. Next we perform image smoothing, to highlight the important features of the retina. We take the weighted sum of the image with a blurred version of the image to remove areas of fairly uniform colour [2]. The areas of

uniform color in the retina are the background and the healthy retina tissue. The areas of non-uniformity in the images consist of the regions containing hard exudates, blood vessels, and the iris. We can see from FIG 5. that these features are made more distinguishable from the background.

Next, we resize the images down to a resolution of 200x200 for practicality and efficiency. As we will be fitting random forest and logistic regression models to these images, we will be using each pixel as a predictor variable. Furthermore, for these two methods we have also applied a grayscale filter [3] to remove any artifacts of the imaging conditions and to further reduce the dimensionality of the data to a single color channel.

Afterwards we perform some feature extraction, as the original dataset had no explanatory variables. We chose features that would highlight anomalies in the images, such as the average distance of a bright pixel to the center of the image as we know that defects in the retina are generally located closer to the center. A comprehensive list of these features can be found in [4]. A few boxplots of these features are shown below for each class.

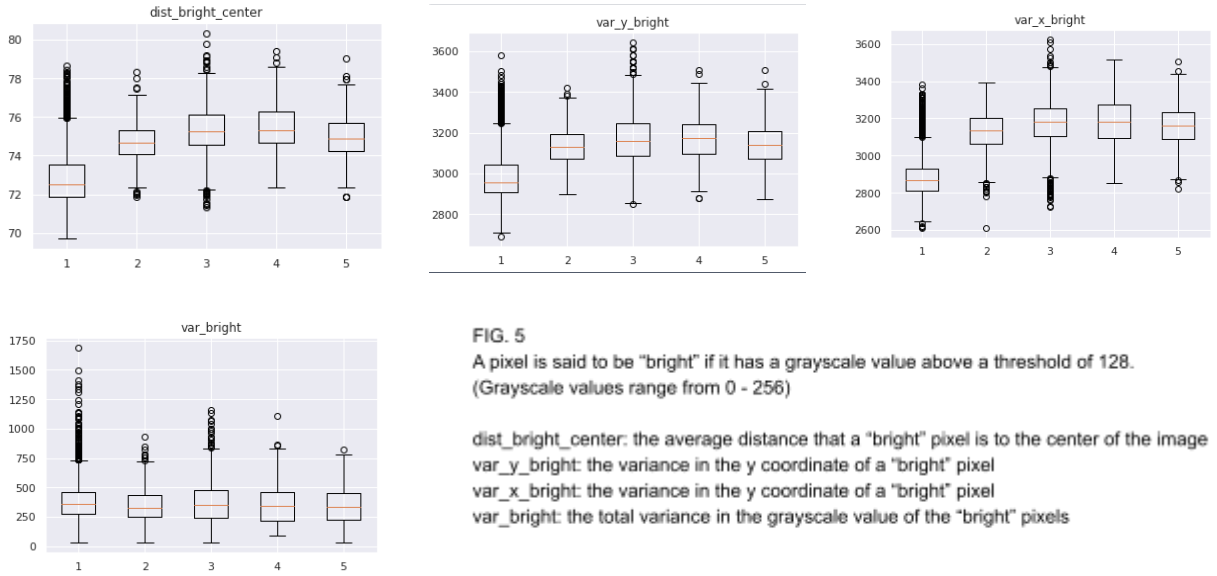

FIG. 5
A pixel is said to be "bright" if it has a grayscale value above a threshold of 128. (Grayscale values range from 0 - 256)

dist_bright_center: the average distance that a "bright" pixel is to the center of the image
var_y_bright: the variance in the y coordinate of a "bright" pixel
var_x_bright: the variance in the y coordinate of a "bright" pixel
var_bright: the total variance in the grayscale value of the "bright" pixels

CNN preprocessing

For our CNN, we perform
the same preprocessing steps above until the resizing where we resize our images to 224x224 instead of 200x200 because of the constraints of our network on the input size.

For our CNN, we also performed test-time augmentation which is done right before loading an image into the network. We chose to randomly flip the images on the vertical and horizontal axis. We also chose to randomly rotate the images. This is done to aid the network to learn features not based on the exact coordinate values but rather on the relative pixel values.

Comparison of models - performance, time

We will explore 3 models for classification of diabetic retinopathy. They are: random forest, logistic regression, and convolutional neural networks.

Random Forest

We first fit a random forest based on each pixel of the images. We used 100 estimators, with 2 minimum samples per split in the model. We used the balanced class weights that were adjusted such that larger weights were assigned to classes with a lower frequency in the dataset and smaller weights to the classes with a higher frequency[7]. This was done to reduce the effects of our unequal class distribution in the data. Then we performed a 4 fold cross validation to obtain the performance metrics.

From the 50% prediction intervals, we notice that our model performs fairly well at predicting class0 (no DR) and class 2 (mild DR). Most of the misclassifications are with the neighbouring classes, which is understandable as the classes are ordinal.

In particular we notice that there is a bit of ambiguity between classifying class1 and class2. However, classes 3 and 4 are very poorly predicted.

Logistic Regression

Next we fit a logistic regression model, first using each pixel as a predictor. This model was also fitted using the balance class weights. Then we performed a 4 fold cross validation to obtain the performance metrics. This model performs extremely poorly, and fails to classify any of the classes to a reasonable degree, further analysis of this will be excluded as we will focus on the next logistic regression model.

The next logistic regression model was fitted from our extracted features of pixel characteristics. This model also performed extremely poorly, but the purpose of this model will be explained in the analysis section.

Convolutional Neural Network (CNN)

Finally, we fit our CNN. A CNN automatically learns features from the data. However, we needed to create a data generator for the CNN. Since our dataset consists of a high number of images from class 0 (no DR), we needed to undersample that class and oversample the other classes proportionally to help the model perform better with the classes with fewer images. The CNN had small prediction intervals and the highest coverage for the 50% and 80%. We notice that the 50% prediction interval had the lowest loss among all our models and can be considered the best model. We define our interval loss in the following manner:

For each image, a loss of 1 is given if the prediction interval does not contain the true label, a loss 1/2 is given if the prediction interval is of size 2 and one of the two predicted classes correspond to the true label, a loss of ⅔ is given if the prediction interval is of size 3 and one of the 3 predicted classes correspond to the true label and so on. We then compute the average of these losses across all the images.

So, our interval losses penalized long prediction intervals and the logistic regression model fared the worst, while the CNN fared the best.
It is also interesting to note the time taken for each model. The preprocessing step was common to all the models and included the feature extraction for logistic regression. The random forest took a few seconds to run while training the CNN to an accuracy of ~80% took an hour. The training accuracy generally increased with time. However, we note that the CNN is much slower than the other models and we can consider a random forest based model for quicker analysis in the future.

A table of comparisons between the different models is shown below:

	50% Prediction Interval - Average Length				
Model	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Random Forest	1.04	1.54	1.42	1.54	1.5
Logistic Regression (Feature Extraction)	4.1	4.3	4.3	4.3	4.3
CNN	1.021	1.245	1.233	1.150	1.240

	80% Prediction Interval- Average Length				
Model	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Random Forest	1.25	2.46	2.37	2.73	2.48
Logistic Regression (Feature Extraction)	4.4	4.8	4.8	4.8	4.8
CNN	1.015	1.230	1.150	1.104	1.210

	50% Prediction Interval - Misclassification Rate				
Model	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Random Forest	0.01	0.400	0.121	0.789	0.875
Logistic Regression (Feature Extraction)	0.865	0.441	0.359	0.187	0.153
CNN	0.003	0.218	0.167	0.121	0.277

	80% Prediction Interval - Misclassification Rate				
Model	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Random Forest	0.010	0.212	0.011	0.606	0.340
Logistic Regression (Feature Extraction)	0.865	0.441	0.359	0.187	0.153
CNN	0.003	0.201	0.108	0.088	0.168

The misclassification rates for the CNN were fairly low except for classifying classes 1 and 4. Random forest had very low misclassification rates for class 0 and 2, which are also the most represented classes in the dataset. Thus this could be an artifact of overfitting, as tree based methods are typically susceptible to.

	50% Prediction Interval - Coverage Rate				
Model	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Random Forest	0.991	0.600	0.879	0.212	0.125
Logistic Regression (Feature Extraction)	0.135	0.559	0.641	0.813	0.847
CNN	0.003	0.218	0.167	0.121	0.277

	80% Prediction Interval - Coverage Rate				
Model	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Random Forest	0.994	0.788	0.989	0.394	0.656
Logistic Regression (Feature Extraction)	0.388	0.918	0.885	0.932	0.973
CNN	0.003	0.201	0.108	0.088	0.168

	Interval Loss	
Model	**50% prediction interval**	**80% prediction interval**
Random Forest	0.356	0.515
Logistic Regression (Feature Extraction)	0.984	0.977
CNN	0.162	0.220

Analysis + interpretation

Logistic regression - pixel wise classification

This model performed relatively poorly, but a heat map shown below of the coefficients provides some insight. The bright regions indicate larger coefficients for the weight in that location of the pixel. It is apparent that for classes 1-4 there are bright regions throughout the center of the images, with class 3 having very noticeably brighter regions near the center of the images. These regions in the retina with DR typically contain anomalies, thus this is at least consistent with our expectations. However for class 4 it appears that there is a decrease in brightness (the weights of the coefficients). Upon inspecting images for the retina scans of class 4, we noticed that a lot of these images contain fewer amounts of hard exudates (than classes 2 and 3), but instead were classified as proliferative DR because of the swelling and clotting of the blood vessels. Although our logistic regression model is unable to classify or interpret the characteristics of the blood vessels, it provided insight on our data.

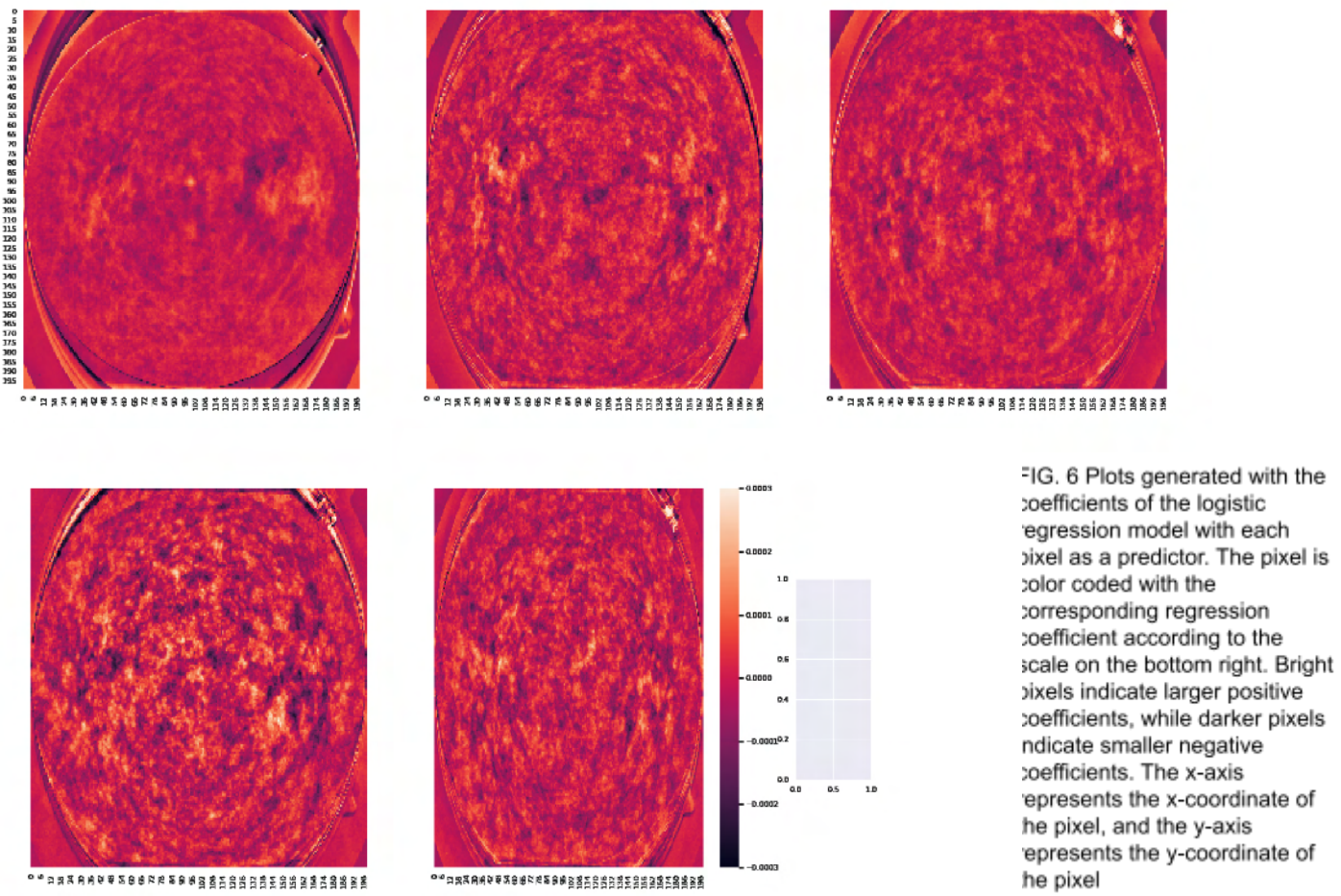

FIG. 6 Plots generated with the coefficients of the logistic regression model with each pixel as a predictor. The pixel is color coded with the corresponding regression coefficient according to the scale on the bottom right. Bright pixels indicate larger positive coefficients, while darker pixels indicate smaller negative coefficients. The x-axis represents the x-coordinate of the pixel, and the y-axis represents the y-coordinate of the pixel

Logistic regression - Feature extraction

There proved to be great difficulty in manually performing feature extraction on images, which is evident in the poor performance of this model. However, this model proves to bring some insight on the characteristics of the images that are important for classifying DR. The most significant features for this model were found to be var_x_bright, var_y_bright, and dist_bright_center. var_x_bright and var_y_bright represent the variance along the x and y coordinates of a “bright” pixel. “Bright” pixels are significant as after our preprocessing, these pixels are often the result of anomalies in the eye such as hard exudates and swollen blood vessels. However, for each retina image there are many bright pixels that highlight features of unimportance such as the outline of the retina, the iris, and healthy blood vessels. Thus the variable “mean_bright” which is the average brightness of the pixels is a poor predictor. However the variance of the location of these bright pixels is seemingly important as they
indicate irregularities in brightness along an axis. The variable dist_bright_center measures the average distance a bright pixel is to the center. We know that irregularities in the eye often persist near the middle of the retina scans, thus it is apparent that this variable would be significant.

Random forest

Surprisingly a pixel-wise classification performed by the random forest model was a relatively good predictor for DR. Unfortunately, performing a pixel-wise classification provided very low interpretability as to how the classifications were made. The splits were made based on the pixel values at each location. However, the interpretation that this model performs well could be due to the fact that anomalies in the eye are typically clustered together and splits in the tree were made under the conditions of neighbouring pixels.

CNN

Compared to the other models, cross-validating with 4-folds and then averaging the predictions of the final softmax layer did not bring about an improvement in our model performance. Other methods of cross-validation like averaging the weights of the neural network is not practical for our task. Another drawback of the CNN was the training time. An epoch (input of 100 images) took about 4 minutes and training a competitive CNN model would take a couple of hours. This meant that one cannot iterate and change the hyperparameters of the CNN too often. Furthermore, CNNs are essentially uninterpretable. This is because we do not know what the features that are learned by the CNN corresponds to.

Conclusion

In conclusion we have fitted 3 different models, and explored the different methodology and results for each. Unfortunately the two models that performed well (CNN and random forest) had very low interpretability. However these models had fairly accurate predictions (CNN by far more accurate). In the practical sense, we see these models as being useful for diagnosing diabetic retinopathy among patients. The tradeoff that our models did not provide much new insight to our understanding of DR appears to be acceptable because this eye condition is already very well understood. The models used for classifying DR could be used to expedite the diagnosis of this condition.

Contribution
Gordon worked on the logistic regression and random forest models while Tarang worked on the CNN. They both worked on the preprocessing, analysis and writing the report.

Appendix

References:
[1] https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/diabetic-retinopathy#:~:text=Diabetic%20retinopathy%20is%20an%20eye,at%20least%20once%20a%20year.

[2] https://docs.opencv.org/3.4/d5/dc4/tutorial_adding_images.html

[3] https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html

[4] feature extraction labels:

Label	Description
mean_pixel	mean pixel value
var_pixel	variance in pixel value
mean_bright	mean pixel value that is bright
var_bright	variance in pixel value that is bright
var_x	variance along the x direction
var_y	variance along the y direction
mean_x_bright	average x coordinate that is bright
mean_y_bright	average y coordinate that is bright
mean_x2y_bright	average x^2 y coordinate that is bright
mean_xy2_bright	average x*y^2 coordinate that is bright
mean_xy_bright	average x*y coordinate that is bright
mean_x2y_bright	average x^2 y coordinate that is bright
var_x_bright	variance of x coordinate that is bright
var_y_bright	variance of y coordinate that is bright
dist_bright_corner	average distance from the corner (0,0) that

	is bright
dist_bright_center	average distance from the center (100,100) that is bright
edge_x	average x coordinate that is bright after applying sobel edge kernel
edge_y	average y coordinate that is bright after applying sobel edge kernel

[5]
Equation for balanced weights: `n_samples / (n_classes * np.bincount(y))`

`Documentation:`
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html