ETC3250: Project (Report)

What's in the data

Data source

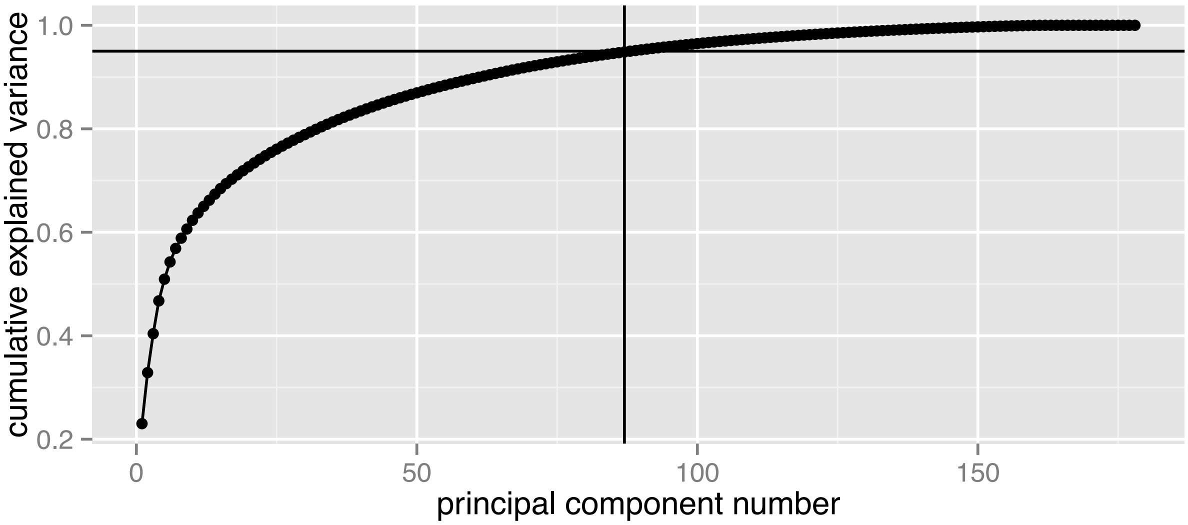
This dataset contains the 242 paintings by Bob Ross for sale from saleoilpaintings.com, which have been re-sized to be 20x20, and classified based on the name of the painting. The primary concern of this dataset is poorly named paintings, which subsequently leads to poorly classified paintings (weakening our classifier).

Painting dimensionality

The dataset dimensionality is high in that for a 20×20 painting there are 400 pixels, each containing a value for red, green, and blue colour. The pixel data makes up $400 \times 3 = 1200$ variables in the wide form dataset. There are far more variables than paintings, and so dimensionality reduction techniques may be useful. It is also notable that there is a severe imbalance in classes, where there are 30 dusk paintings but only 5 oval paintings in training set.

Pixel similarity

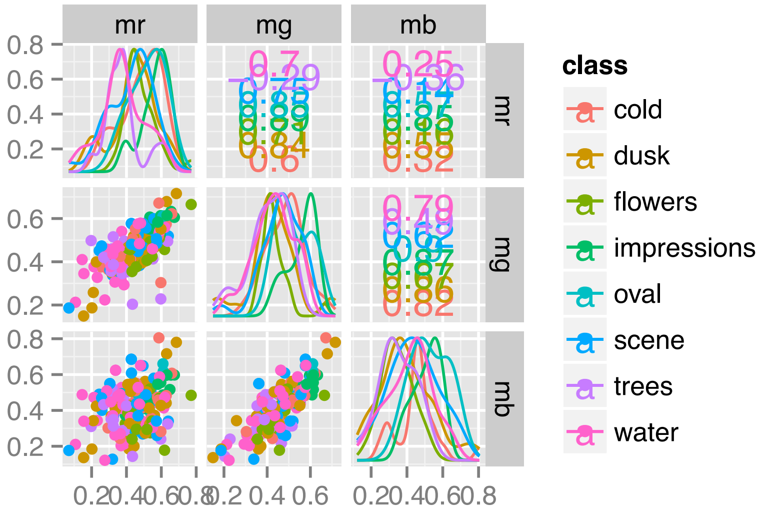
With a hierarchial data structure such as an image, we expect nearby pixels to exhibit a similar colour, and so for example, r1 should be similar to r2 as they are neighbouring pixels. This is evident through principal component analysis, where the screeplot indicates that the most of the variance is explained in the first few principal components.



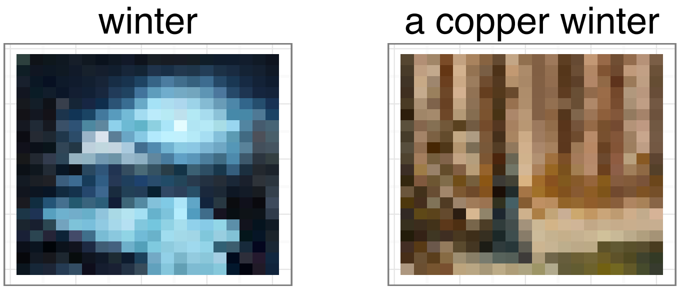
The line in the above figure indicates that 95% of the variation is explained by the first 87 principal components, far less than the 1200 colour variables that are provided.

Class colours

Taking the average pixel colours from each painting gives the following scatterplot matrix by colour.



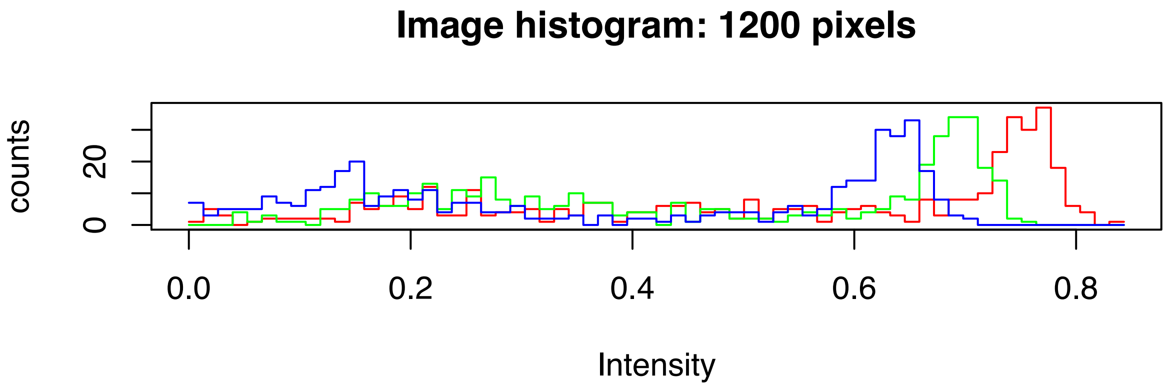
Interestingly we can see that many classes have more than one peak in the density of average colours. This is in part due to the low sample size, but is also indicative of how paintings in the same class can have different styles. For example, the following two paintings are classified as “cold” paintings, however one has much more of a warm style (and accordingly contains much more red making it visually similar to dusk paintings).



Similar extreme differences exist within the flowers, impressions, dusk and water painting classes.

Pixel colour histogram

The histograms of painting classes reveal notable differences in oval paintings as compared to other classes. Converting the images into the EBImage format makes for simpler image analysis and manipulation. Ovals have large areas of similarly coloured pixels, and so there are strong peaks in all colour layers.



Kaggle and Objectives

For the purposes of the Kaggle competition, we are attempting to classify paintings based on the pixel information for each painting. The objectives of this project is to build a strong classifier for painting types, test whether paintings can be identified by their picture colours, whilst identifying the paintings that are poorly classified and why.

Building a classifier

Contextual investigation

Before building any classifier I investigated any additional contextual information that would support painting classification. This involved reading previous research done on Bob Ross paintings and image manipulation techniques to give ideas on how to create new and more valuable features from the paintings data. The key findings were that to add value to the data, we need to create new features that represent the hierarchial structure of images.

Baseline classifier

Based on the dimensionality of the dataset, my baseline classifier was a random forest without any feature engineering. This performed surprisingly well, correctly classifying over half of the test set of paintings. Due to its effectiveness, I opted to build upon this random forest with new features.

Feature engineering

Luminosity

The first feature implemented was converting the colour images to grayscale using the luminosity method as described by John D. Cook. This method was chosen as it accounts for human perception, which is highly influential in artistic works. The formula for this is as follows:

$$\text{lum} = 0.21R + 0.72G + 0.07B$$

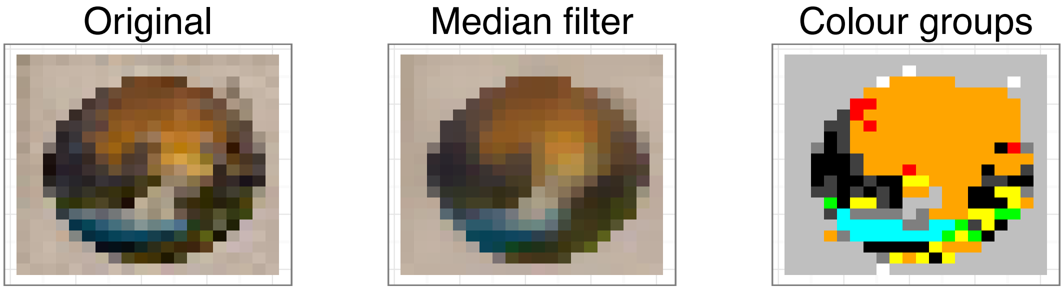
This fails to add much to the random forest as it is simply a linear combination of pre-existing colours and does not provide additional hierarchial information. It is however useful for quickly understanding the data and reducing the dimensionality.

Converting RGB to HSV

By transforming the colours in this way we are able to make a more immediate approximation of the colour by looking at the hue. In doing this conversion, more value is given to vibrance and colour saturation, as opposed to the colour itself. For a random forest, it is helpful to include the saturation and vibrance information as these add new information/interpretations to the model. Hue is better used for classifying the colour of each pixel, visually in accordance to the above reference image.

Median filter

Using the EBImage package to create a median filter on the paintings improves the visual cleanliness of the images. The downsizing process introduces a lot of noise which in an analytical sense is an overfitting of the actual painting class. Using a median filter is a useful pre-processing task that improves the grouping of similar colours in paintings. We can also then define ranges of HSV values to match to a group of colours, to further group the colours together. Further tweaking can improve the accuracy of the colour groupings.



Oval separation

As seen in the image groupings for the above oval paintings, all of the pixels in the corner have been grouped into the same colour. So a metric for separating oval paintings can include the 6 corner-most pixels from each corner. Oval paintings should have more corner pixels separated into the same group, and so a higher number should result.

Enforcing predictions to be proportional to the training classes

This was implemented to improve upon the unbalanced class issue as often the random forest would over-predict certain classes. Also for the purposes of the Kaggle competition, my testing indicated that the training and test sets were separated in a stratified manner. Using a greedy algorithm from the probability output of the random forest, I allocated classes based on the most confident predictions down to the least confident predictions. When the most preferred class was full, I made the prediction select the second best option.

Results of the classifier

Performance of classifier

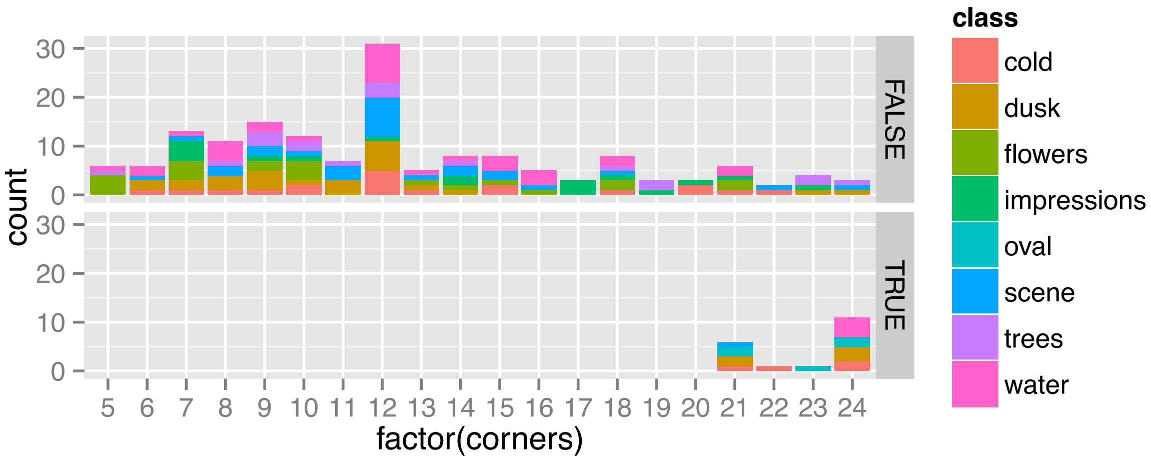
The Kaggle scores of the random forest with additional features included often did not improve upon the basic random forest from training set. The scores from my classifiers were around 50% accurate when submitted to Kaggle. Upon looking at the predictions that were outputted, the additional hierarchial features exacerbated the class imbalance problem, causing most predictions to be a common class like water.

Misclassified ovals

The clearest cause for these problems is the misclassification of paintings, most evident in oval paintings. Using the corner mode algorithm described above, we can see that many paintings that are not classified as oval paintings actually do exhibit the oval feature.

```
##      id                name class corners oval
## 12   13     auroras-dance-1  dusk      21 TRUE
## 29   33            seascape water      24 TRUE
## 47   56    fishermans-cabin water      24 TRUE
## 66   84 haven-in-the-valley scene      21 TRUE
## 87  119     auroras-dance-2  dusk      21 TRUE
## 90  123      lonely-retreat  cold      22 TRUE
## 93  126            winter-2  cold      21 TRUE
## 116 164         winter-lace  cold      24 TRUE
## 119 168     winter-paradise  cold      24 TRUE
## 121 170     twilight-beauty  dusk      24 TRUE
## 140 195  hidden-winter-moon  dusk      24 TRUE
## 147 205      babbling-brook water      24 TRUE
## 166 227 fishermans-paradise water      24 TRUE
## 176 240        pinky-sunset  dusk      24 TRUE
```

Having manually went through to ‘correctly’ classify paintings as oval or not, 19 paintings in the training set are ovals. However only 5 are classified as oval paintings. This is very detrimental to our classification of oval paintings. Most of the paintings that the corner algorithm incorrectly identified (using 21 as the minimum) consist of that same colour across the entire painting. So an improvement would be to compare the mode colour of the corners with the mode colour of the center before deciding the painting to be oval.



Poor data source

This incorrect classification continues into the other classes, and is a result of the data source. Some of these paintings from the website are not named at all, incorrectly spelt and some do not represent the class at all. There are also some paintings which do not belong to any class, so they are difficult to classify.

Conclusions

From the performance of the classifiers (with consideration to the volume of unusually classified paintings), achieving an above 50% accurate predictions makes clear that the class of paintings can be determined by its pixel colours. Further improvements to the classifier involve introducing a sense of hierarchial and contextual structure into the paintings data, as this is is not implied when directly applying a random forest. This project illustrated the importance of quality over quantity for classification problems, as the misclassified data has impeded on our model's performance. This suggests that the name of the paintings doesn't necessarily imply a certain colouring theme in the painting.

There are many image analysis techniques which have been employed in creating this classifier, which has led me to investigate different ways of representing colour (HSV). In a more detailed (higher dimensional) data set, it may be useful to rotate and flip images to create more training data. This helps with identifying features in the paintings (such as the existence of a mountain or lake) that would help build a stronger classifier. Quirks of human perception such as increased sensitivity to green colours (as used in luminosity formula) and the application of a median filter to smooth noisy images allows for a better representation of a typical painting in each class.

Bonus marks: Kaggle Classification Accuracy (Multiclass)

The score on the Kaggle leaderboard suggests that the score is a form of "categorisation accuracy", although no details of the metric is available from Kaggle. To find the formula for the measure, I attempted to get only one predicted class correct (resulting in me finding the error in the Kaggle comparison csv). In the original monba competition, one correct entry increases the score by 1/63, and this score is not impacted by any non-linear interactions with other correctly predicted classes (or non-sensical input). As a result, the accuracy measure can be deduced as:

$$\text{Score} = \frac{\text{number of correctly classified paintings}}{\text{total number of test paintings}}$$

This formula continues to be consistent in the monba2 competition, although it is split by the public and private leaderboards. The public leaderboard is based on 49% of the data (31 paintings), and so the score is calculated on public and private split.