Neural Networks

Detailed Report

Team III

Name	Section	Bench No.
Essam Wisam	2	1
Iten Emad ElDin	1	19
Mohamed Saad	2	14
Radwa Ahmed	1	30

Supervised by

Dr. AbdelMoniem Bayoumi

Eng. Nesma Refaei

May 2022

Table of Contents

Literature Review

Gender detection using machine learning has developed to a decent extent over the past decade. We considered several papers while implementing our project. The following are the papers that really stood out to us and the ones we frequented a lot in our trials.

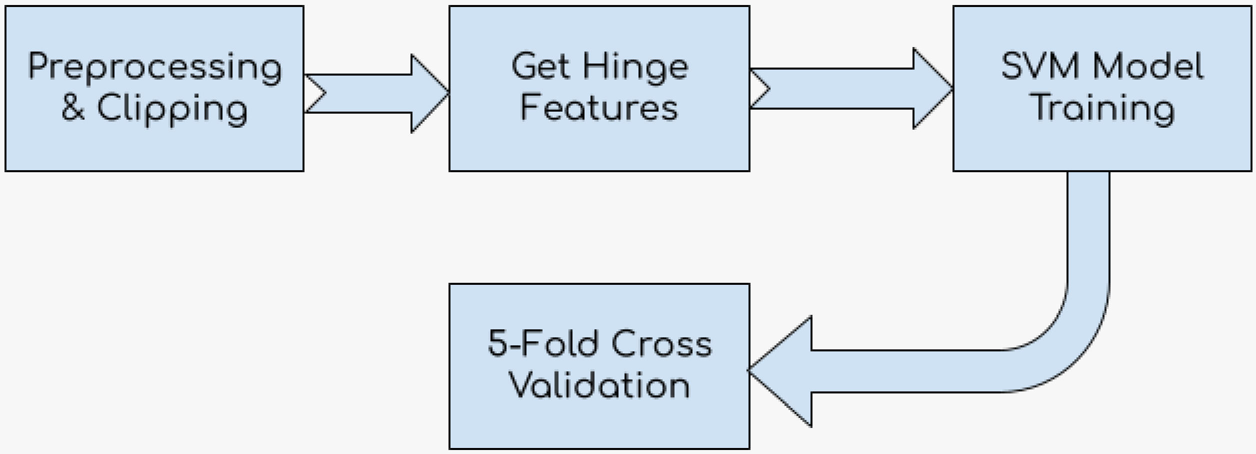
1. Improving handwriting based gender classification using ensemble classifiers, We consider this paper to be our first main reference since it has compared a wide range of features and their combinations. It included features such as local binary patterns (LBP), histogram of oriented gradients (HOG), statistics computed from gray level co-occurrence matrices (GLCM) and features extracted through segmentation-based fractal texture analysis (SFTA).

   Regarding classification models, it employed artificial neural networks (ANN), support vector machine (SVM), nearest neighbor classifier (NN), decision trees (DT) and random forests (RF). Classifiers are then combined using bagging, voting and stacking techniques to enhance the overall system performance.

2. Handwriting Based Gender Classification Using COLD and Hinge Features, As apparent from its name, this paper mainly employs SVM models for each of the two feature sets and takes their maximum as the predicted classification.
3. Gender classification from offline multi-script handwriting images using oriented Basic Image Features (oBIFs). Last but not least, this paper produces oBIFs histograms and oBIFs columns histograms and uses an SVM model to classify. It is noteworthy to say that it utilized the ICDAR 2013, ICDAR 2015 and ICFHR 2016 datasets.

From all the previous papers and many others, we concluded that using SVM was one of the most popular choices for a model used in gender classification problems.

Project Pipeline



After several research attempts and multiple trials in terms of models and features choice (as will be demonstrated later), the final project pipeline we propose is as follows:

1. Pass the training set images to the preprocessing model by removing shadows and clipping the area that has the handwriting.
2. Extract the Hinge features from these images through resizing images, dividing them into blocks, calculating magnitude, angle of gradient for each block, and calculating the histogram for n bins.
3. Pass the resulting features to the support vector machine (SVM) model to start training it using a RBF kernel.
4. Use 200 iterations of 5-fold Cross Validation to get a relatively accurate representation of the model's accuracy and do a proper performance analysis.

Preprocessing Module

It is a pretty straightforward process (that isn't so straightforward to implement): convert the image to grayscale, resize it, remove shadows from the image, convert it to binary, remove noise and finally clip the image to the part containing the handwriting.

Feature Extraction & Selection

After too much analysis (as demonstrated below). We ended up choosing hinge features without combining it with another feature. We also used the feature vector directly without applying PCA to reduce dimensions.

The method to extract Hinge features goes as follows:

1. A medial filter is applied to reduce the noise.
2. A threshold is applied to convert the img to a binary form.
3. The contours are extracted and sorted by the area.
4. Filter the small contours.
5. Loop over all the contours and get the angles between points.
6. Obtain a normalized histogram of the angles.

Model Selection & Training

We selected an SVM model with a RBF kernel with the regularization parameter set as 6.7.

Performance Analysis

For performance analysis we relied on 5-fold cross-validation under 200 randomizations of the model to reach a final accuracy of about 90.57% with 25s of runtime. For more, check the formal comparison below.

Additional Optional Modules

We can combine more than one feature with the Hinge feature to get higher accuracy by applying more visualization to understand more how it works . The feature chef module was responsible for that (as will be demonstrated below.)

Formal Comparison of Features & Models

We started choosing different features to work with. They are GLCM, HOG, LBP, Fractal, Hinge and COLD.

The features gave us different results depending on some parameters and classifiers. Some features gave us large feature vectors which may cause problems, so sometimes it was best to use PCA to reduce large dimensions of features.

To set out the formal comparison we will begin by laying out a preface for each of the features, following that with the method and visualizations then finally topping that off with the testing results.

1 - GLCM Features

Preface:

The Gray-Level Co-Occurrence Matrix known as GLCM, is a matrix that considers the spatial relationship of pixels. GLCM functions describe the texture of an image by calculating how often pairs of pixels with certain values and in a designated spatial relationship occur in an image and then extracting statistics from this matrix. For example:

- Contrast: Measures the local variations in the GLCM matrix.
- Correlation: Measures the joint probability of occurrence of the specified pixel pairs.
- Energy: Provides the sum of squared elements in the GLCM. Also known as uniformity or the angular second moment.
- Homogeneity: Measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.
- Entropy: Measures of the uncertainty associated with pixels.

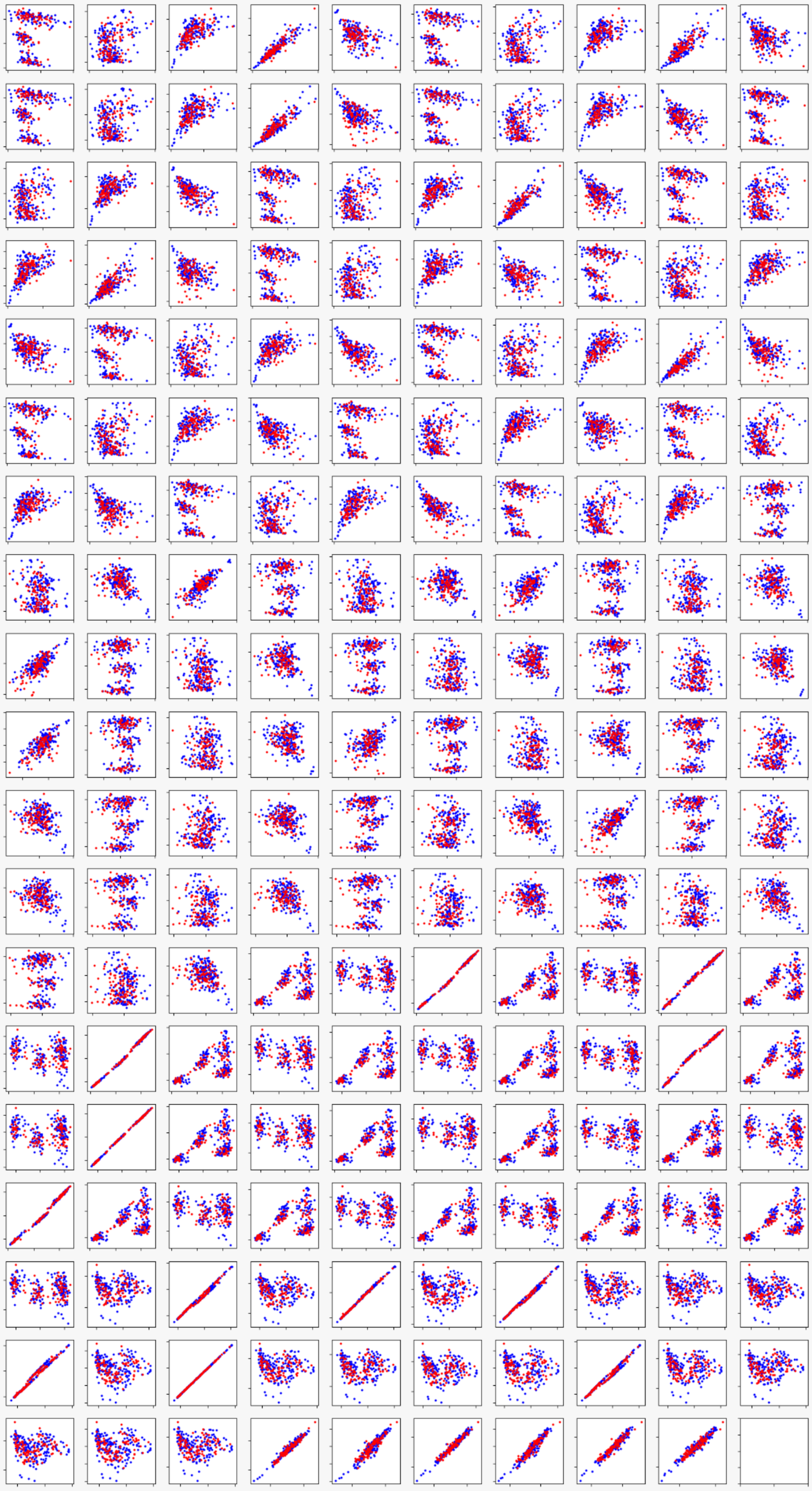
SNo.	Feature	Computational details
1.	Contrast	$\sum_{i,j=0}^{N-1} P_{i,j} = (i,j)^2$
2.	Correlation	$\sum_{i,j=0}^{N-1} P_{i,j}\left[\frac{(i-\mu_i)(j-\mu_j)}{(\sqrt{(\sigma_i^2)(\sigma_j^2)})}\right]$
3.	Homogenity	$\sum_{i,j=0}^{N-1} P_{i,j} = 0\frac{P_{i,j}}{1+(i-j)^2}$
4.	Entropy	$\sum_{i,j=0}^{N-1} P_{i,j}(-In P_{i,j})$
5.	Energy	$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[P(i,j)]^2$

Method:

For each of the following features, we get them at 4 offsets: 0, 45, 90 & 135 at a unit distance.

Visualization:

The combination pairs each of 2 dimensions out of 20 dimensions.



Model Testing:

Using 5 dimensions extracted from the following 5 GLCM features: Contrast, Correlation, Homogeneity, Entropy & Energy, measured at 4 offsets: 0, 45, 90 & 135 at a unit distance, and take the average of the outputs at these offsets for each feature.

Model	Linear SVM		Non Linear SVM		AdaBoost	
Dataset	CMP23	CMP23 + ICDAR	CMP23	CMP23 + ICDAR	CMP23	CMP23 + ICDAR
Accuracy	%37.363	%62.637	%64.835	%62.637	%61.538	%45.055

Model	Random Forests		Random Forest with Cross Validation		XGBoost	
Dataset	CMP23	CMP23 + ICDAR	CMP23	CMP23 + ICDAR	CMP23	CMP23 + ICDAR
Accuracy	%57.143	%50.549	%56.115	%51.126	%54.945	%57.143

Using the 20 dimensions extracted from the following 5 GLCM features: Contrast, Correlation, Homogeneity, Entropy & Energy. Each is measured at 4 offsets: 0, 45, 90 & 135 at a unit distance.

Model	Linear SVM		Non Linear SVM		AdaBoost	
Dataset	CMP23	CMP23 + ICDAR	CMP23	CMP23 + ICDAR	CMP23	CMP23 + ICDAR
Accuracy	%64.835	%34.066	%64.835	%67.033	%57.143	%58.241

Model	Random Forests		Random Forest with Cross Validation		XGBoost	
Dataset	CMP23	CMP23 + ICDAR	CMP23	CMP23 + ICDAR	CMP23	CMP23 + ICDAR
Accuracy	%51.126	%48.352	%65.297	%50.002	%56.044	%46.154

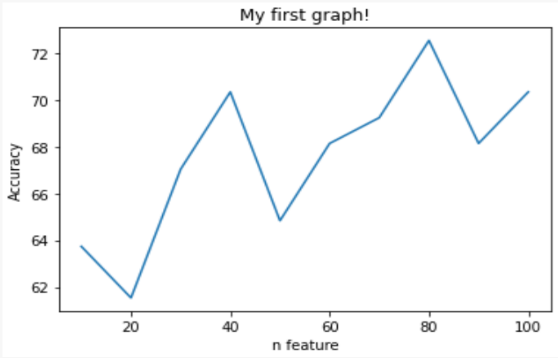
2 - HOG Features

Method & Preface:

It starts by resizing the images, then divides them into blocks, then calculates the magnitude , angle of gradient, for each block, and calculates the histogram for n bins (usually 9).

Visualization:

Hog returns about 14k feature vector, it depends on the size of the images & its hyperparameters, so it will be hard to visualize the feature vector, so the following graph shows the accuracy results using PCA:



Model Testing:

Model	Adaboost	XGBoost	Linear SVM	Nonlinear SVM	Random Forest	Random Forest + CV	Random Forest + PCA
Accuracy	66%	55%	67%	66%	62%	65%	65-72%

Combining Features with HOG using Random Forest:

Features	GLCM	LBP	COLD	HINGE
Accuracy	51.64%	61.5%	70%-74%	60%-70%

3 - LBP Features

Preface:

Local binary pattern (LBP) is one of the most famous features when it comes to texture analysis. It tries to deduce repeating patterns in the image (encapsulated in the handwriting) by comparing the center pixel to the pixels in its neighborhood.

Method:

In its primitive form It works by calculating an 8-bit number at each pixel and constructing a histogram from that value. That is, for each center pixel we check if each of the neighboring pixels is larger or smaller and get an 8-bit binary number as a result from which we construct a histogram.

Model Testing:

Model	SVM with 5-Fold Cross Validation
Dataset	CMP23
Accuracy	63%

Model	Random forest with 5-Fold Cross Validation
Dataset	CMP23
Accuracy	65%

Model	XGBoost with 5-Fold Cross Validation
Dataset	CMP23
Accuracy	64%

4 - Fractal Features

Preface:

The fractal features are about computing the fractal dimension of a given image which is in some sense a measure of the amount of detail in it. The more change we see as we zoom in the higher the fractal dimension is. It has a closed formula for mathematical objects but can still be approximated using algorithms given a binary image. Like the three following features, our motivation for trying out this one was that it's being used in the paper we're following. The paper reports 55% accuracy using fractal features (it was the fourth best feature out of the four the paper considered.)

The output of the method is a 12D feature vector (comprising fractal dimensions and mean values of the image after applying two-threshold binary decomposition. 6 times given 3 thresholds and getting the mean value from each. The paper used one more threshold value and included the pixel count for each to arrive at a 24D feature vector but this is clearly redundant.

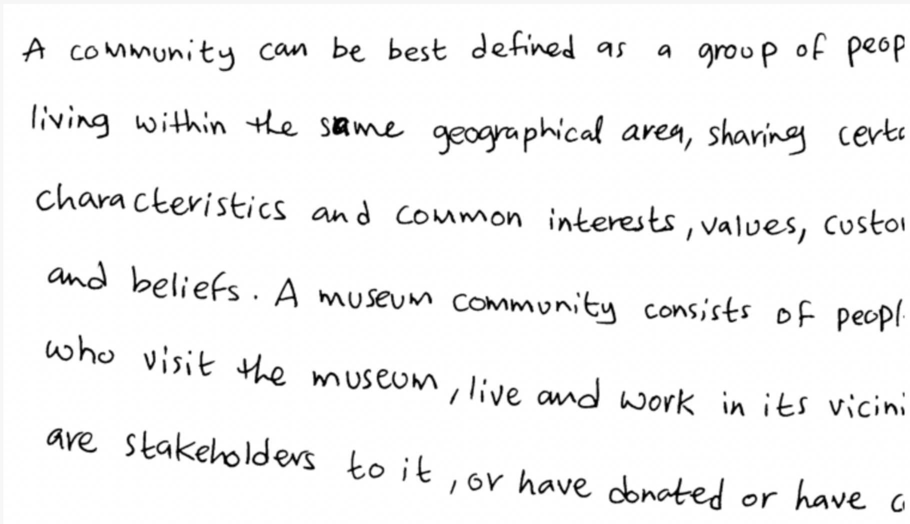
Method:

Minkowski–Bouligand_dimension

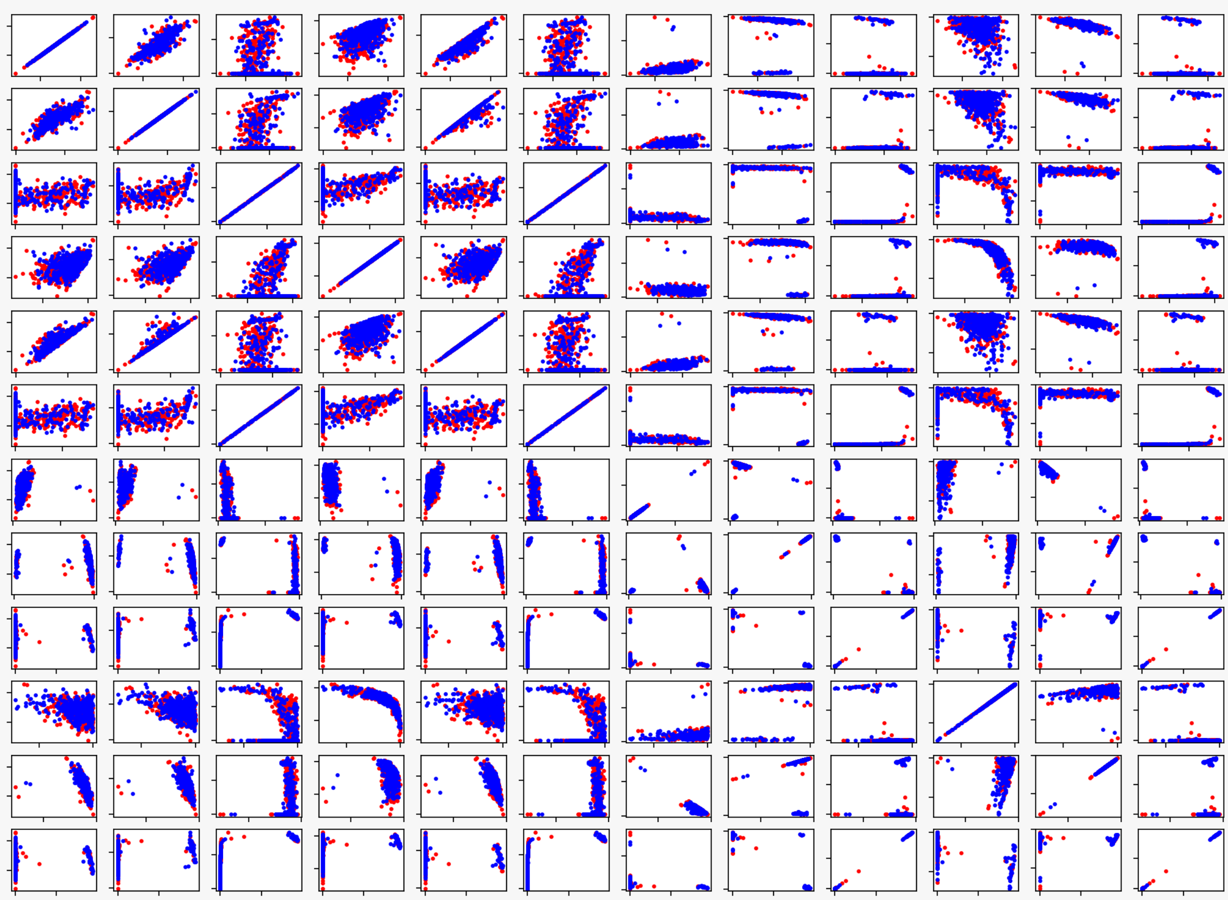
1. Divide the image into four slices then count the number of slices that are not all black or white (including detail).
2. Divide each slice into four slices again and count again.
3. Keep going until the image can't be divided any further.
4. Perform linear regression on the log of the counts versus the number of size of the slice and use it to estimate the count for very small slice sizes (i.e. for which log(size)=0).
5. Return the result as the fractal dimension

Visualization:

Thresholding of a random image:

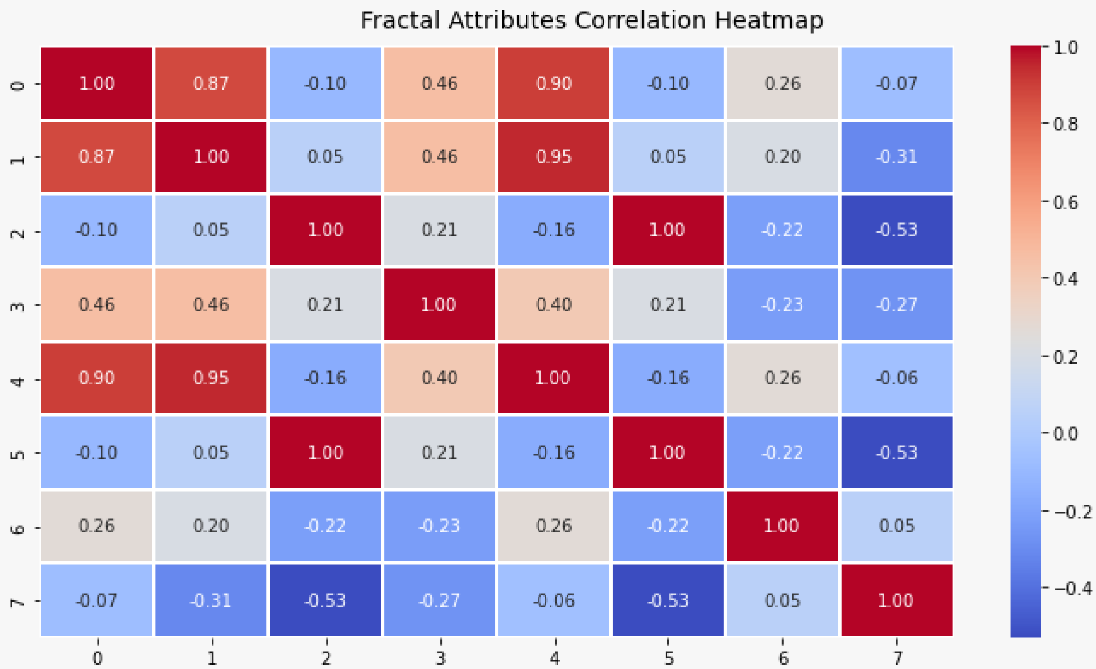


Plotting all possible pairs of features (zoom in):

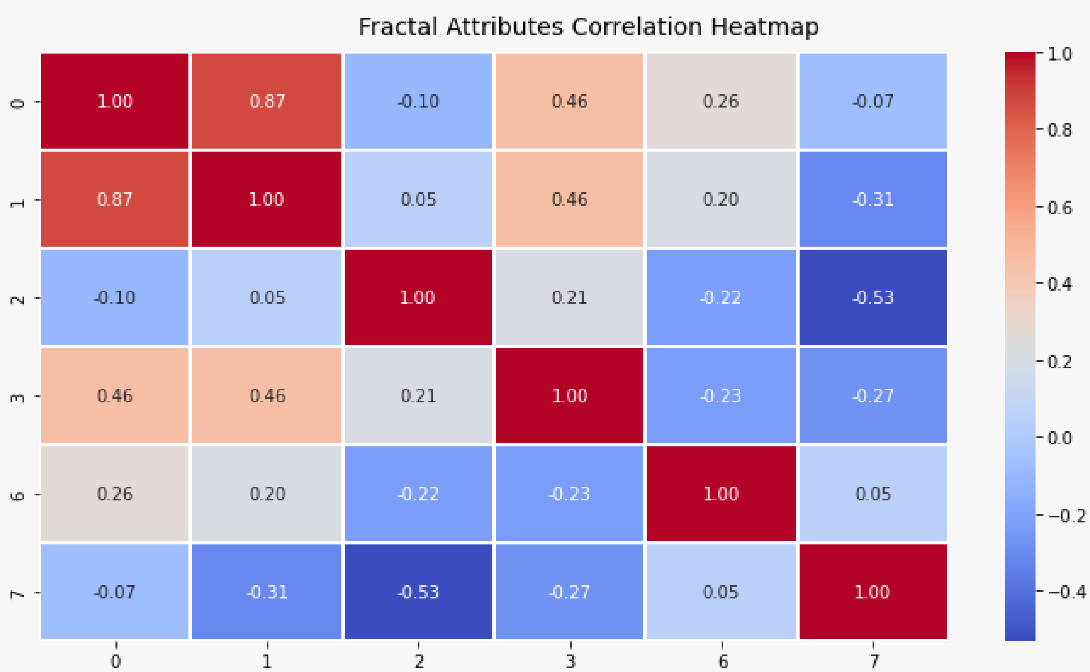


It’s clear that no pair is even close in making the two classes separable (let alone linearly separable.)

Fractal Features Correlation Heat Map:



Fractal Features Correlation Heat Map After Removing Highly Correlated Features:



Model Testing:

Model	Random Forest with 5-Fold Cross Validation	
Dataset	CMP23	CMP23 + ICDAR
Accuracy	59%	55%

Model	XGBoost with 5-Fold Cross Validation	
Dataset	CMP23	CMP23 + ICDAR
Accuracy	54%	51%

Both XGBoost and Random Forest were set to grow 300 trees.

Model	SVM	NN
Accuracy	53.3%	52.2%

The SVM used a Radial Basis Function kernel with the regularization parameter set as 6 and the neural network used two hidden layers with Relu activations and sizes of 40% and 20% of the input respectively.

As can be seen fractal features are doing relatively poorly relative to other features. Our first insight was indeed that info about how "thorny" is the handwriting won't help so much discriminate between males and females. The paper we originally followed as well showed insignificant accuracies for fractal features.

5 - COLD Features

Preface & Method:

The COLD (cloud of line distribution) features comprise a histogram of angles of line segments that join dominant edge points weighted by the length of the line segment.

It comprises 2 simple steps:

1 - Join each dominant edge point to the closest one to form a line segment.

2 - Measure the length and the angle of the formed line segment and increase the corresponding bin in the angles histogram.

Cursive or small handwriting would be expected to have low values in the histogram (correspond to smaller line segments.) The angles of the line segment are also a characteristic that differs based on how the letters are drawn which make this an effective feature for handwritten recognition.

Model Testing:

We used 200 randomizations of the dataset with 5-fold cross-validation to obtain the following results:

Model	SVM	Random Forest	XGBoost	Neural Network
Accuracy	80%	73%	76%	79% (Only 5 CVs)

SVM consistently takes the shortest time among all, with NNs taking the most time.

6 - Hinge Features

Preface:

It is a contour-based feature that tries to capture the curvature of the font.

Method:

7. A medial filter is applied to reduce the noise.
8. A threshold is applied to convert the img to a binary form.
9. The contours are extracted and sorted by the area.
10. Filter the small contours.
11. Loop over all the contours and get the angles between points.
12. Obtain a normalized histogram of the angles.

Model Testing:

Most of the model testing was run over many iterations then the average was taken.

Model	SVM	
Dataset	CMP23 (5-Fold Cross Validation)	CMP23 + ICDAR (10-Fold Cross Validation)
Accuracy	89%	77.88**%

Model	Random Forest	
Dataset	CMP23 (5-Fold Cross Validation)	CMP23 + ICDAR (Random runs)
Accuracy	81.04%	67.03**%

Model	Gradient Boosting
Dataset	CMP23 (5-Fold Cross Validation)
Accuracy	82.08%

Model	Ada Boost
Dataset	CMP23 (5-Fold Cross Validation)
Accuracy	73.00%

We found that the SVM was the prominent one so some tests were run to select the best parameters, from that we found that the Radial basis function kernel with the Regularization parameter(C) = 7.6, and below are some of C values compared to the accuracy (from a random run).

1 76.92 | 7 84.61

2 80.21 | 8 84.61

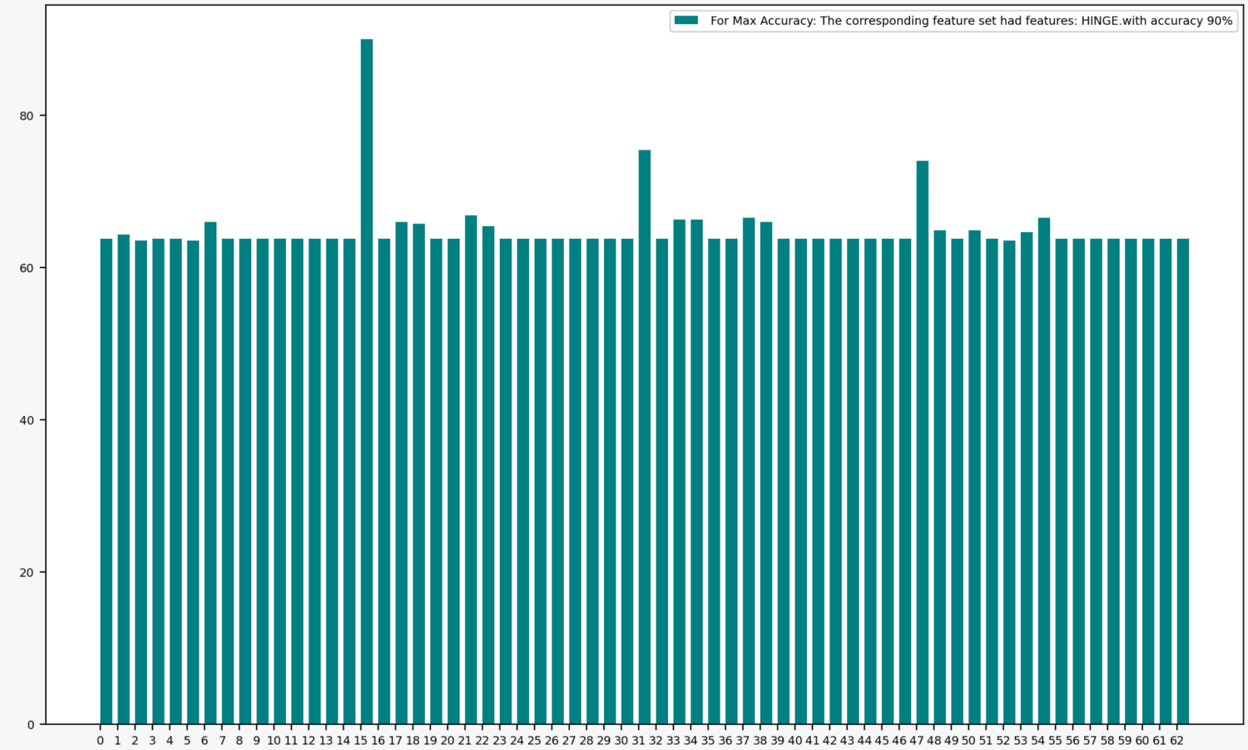
3 80.21 | 9 84.61

4 82.41 | 10 84.61

5 82.41

6 83.51

It's obvious at this point that HINGE features are outperforming all others and that they are best synergizing with an SVM model. However, naturally combining different features can lead to very different results. Because we have 6 features there are 64 possible combinations. To know which is the best we tried all of them on a SVM and produced the following result:



Each integer corresponds to a different combination out of the 64-1. In particular, each bit in any of numbers decides if the particular type of features was part of the combination (COLD | HINGE | GLCM | FRACTAL | HOG | LBP). It's obvious that HINGE and HINGE alone provided the best accuracy (and with a good margin.)

The best model for HINGE was SVM which is why we settled on that as well for the model in our pipeline.

Enhancements & Future Work

We can use more neural network layers to improve accuracy with backward propagation. We also can increase the size of the dataset by combining it with the ICDAR 2013, ICDAR 2015 and ICFHR 2016 datasets. Moreover, we can use bagging, voting or stacking techniques with the SVM model and involve different models together.

Workload Distribution

Initially, we set out the project such that Saad who was concerned with doing the preprocessing, Iten and Radwa for exploring research papers and studying their features and Essam for initiating the model.

After settling on our first research paper which had features HoG, LBP, Fractal Features, GLCM. Each of us considered one of them and worked on its implementations, training and visualizations. Later in the project Saad introduced COLD and HINGE features, Iten worked on the Submission module, Essam worked on major refactoring and the feature chef module and Radwa worked on combining features as well. The fact that one of us worked on specific features didn't mean that they were the only one testing it. This is something that anyone would do and we'd share the results on our online group.