OPTICAL CHARACTER RECOGNITION

PATTERN RECOGNITION(EEL 6825)
SPRING 2016
UNIVERSITY OF FLORIDA

SARAVANAN SETTY
UFID : 3329 9221

OCR PROBLEM

- Convert images of text to an machine readable text document format.
- Image Types : Scanned Text Documents, Computer Generated Text Screenshots,
- Text Types : Computer Generated Text, Handwritten Text
- Handwritten text : Further classified into cursive and block

OCR APPLICATIONS

- Problems with physical documents : fragile, uses a lot of space, searching particular piece of information can be hard.
- Using a scanned image, fragility and space problem is solved, but others remain.
- Introduces new problem of taking lot of space in memory.
- OCR, extracts only the text information from image and stores it in machine readable format and hence search is possible.

STEPS OF OCR

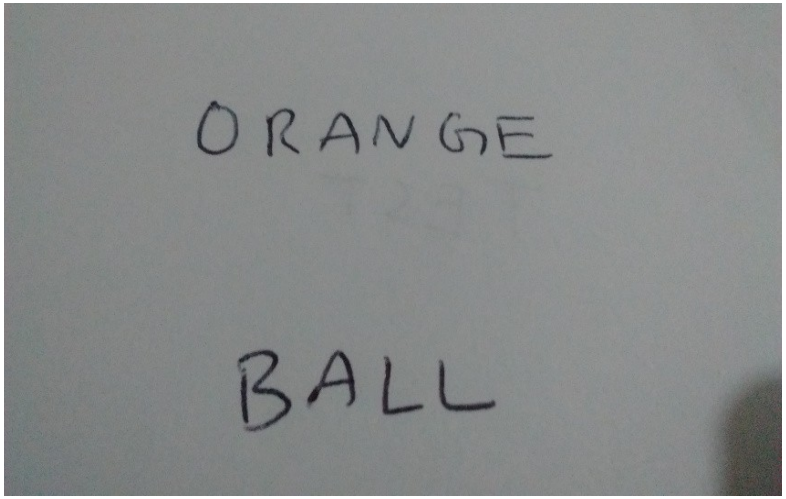
- There are 6 six steps involved in the process of OCR.
- They are : Obtaining Raw Images, Preprocessing, Segmentation, Feature Extraction, Classification, Post processing

STEP 1 : Obtaining Raw Images

- Sources : Screenshots of computer generated text, Images of handwritten text.

APPLES ARE RED

ORANGE IS ORANGE

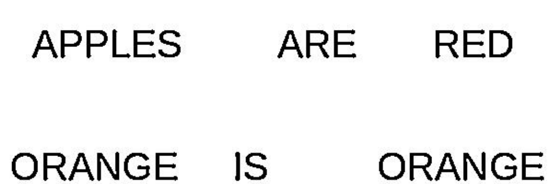


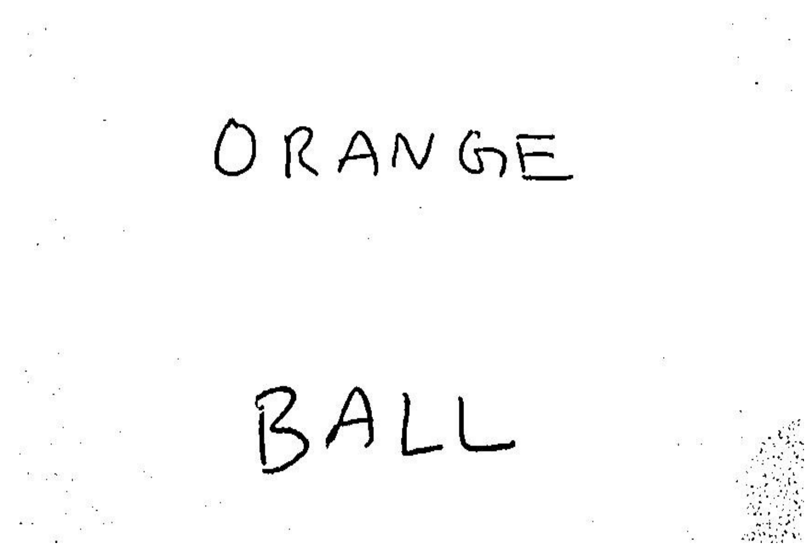
STEP 2 : Preprocessing

- The document is binarized by using adaptive thresholding.
- Adaptive thresholding : Different thresholds for different region of image.
- Works even when different areas of image have different levels of illumination.

Preprocessing : Thresholding on Sample Image

- In case of image, salt and pepper noise is present which needs to be removed.
- Image has been negated to show noise, clearly.



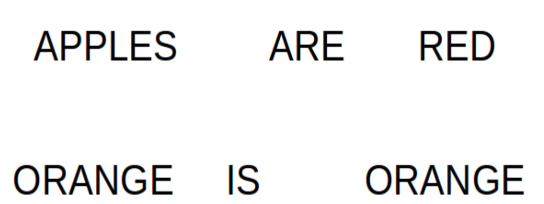


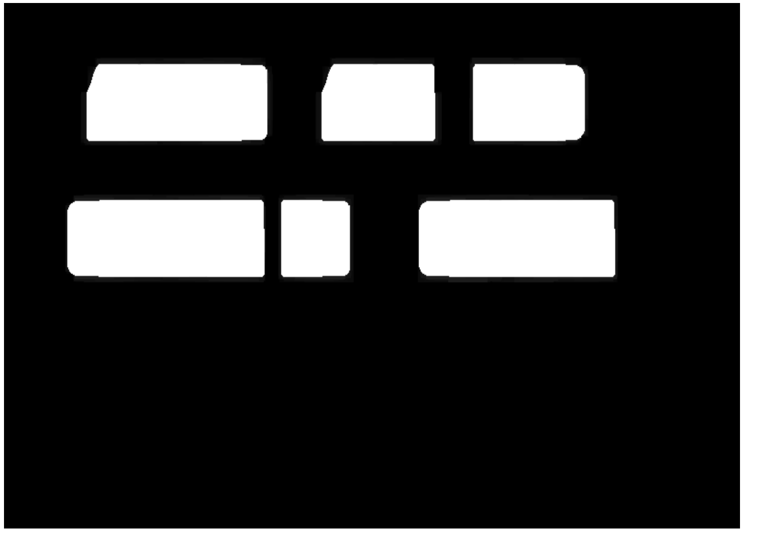
Preprocessing : Removing Noise

- Median Blurring can be used to remove the noise.
- A filter is applied and each pixel is replaced by the median of the pixel intensities present in the filter.

STEP 3 : SEGMENTATION

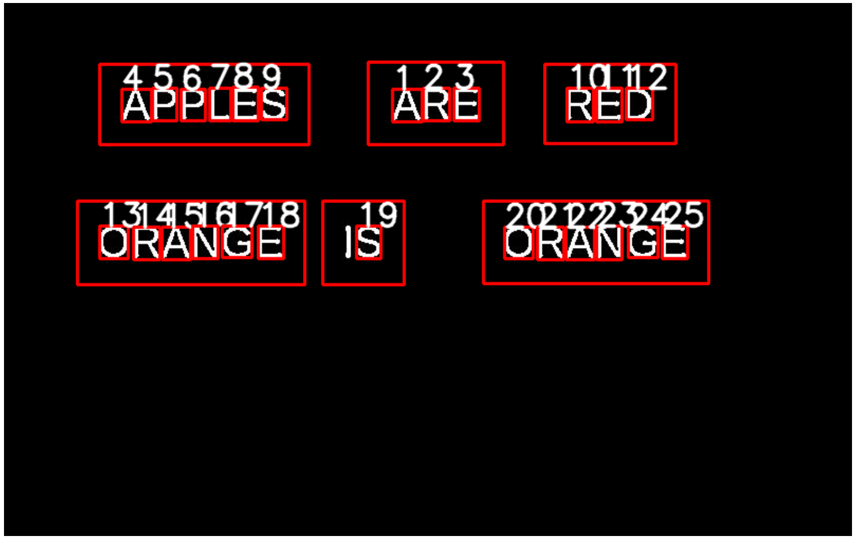
- Here we identify the text regions in the image first.
- First, we need to detect each word in the image.
- For this, dilation is applied so that each word forms a blob.





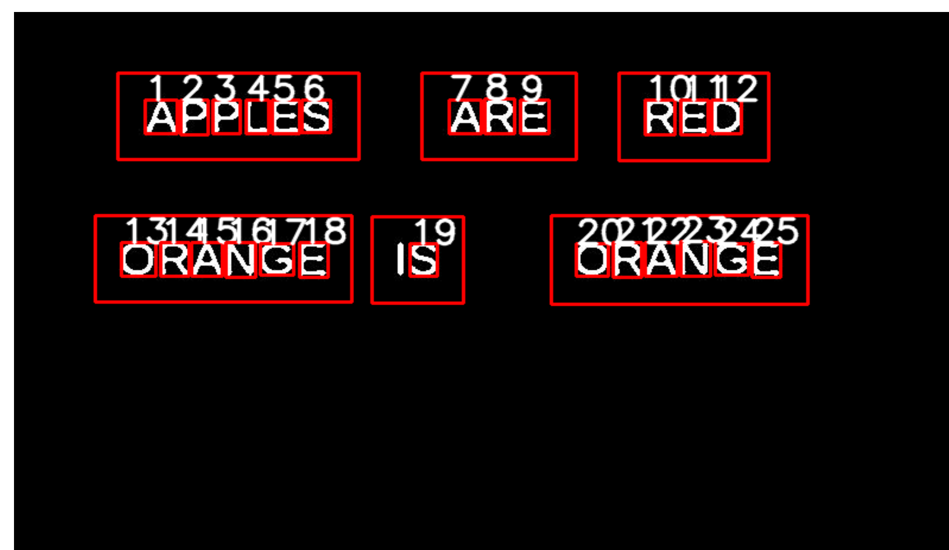
SEGMENTATION : Bounding Boxes and Order

- Maximally Stable Extremal Regions(MSER) is used to locate the text blobs.
- A bounding box is created around each word.
- The words are sorted from top to bottom and then left to right. Characters inside a box are sorted in x-coordinate order.



Segmentation : Fixing the Order

- Order is messed up because even though “ARE” and “APPLES” appear on same line, “ARE” is higher.
- Solution : Use custom sort which allows equivalence for approximately same height.

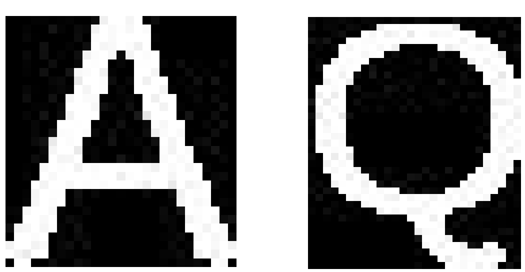


Segmentation : Detecting Characters

- Once the word rectangles are isolated, MSER is applied again on cropped portion of image.
- This allows us to isolate characters in each word.

STEP 4 : Feature Extraction

- In this step, image corresponding to a character is taken as input and features are extracted from it which are used for classification.
- Features used are : Mean X Value, Mean Y Value, Number of black islands, Grid vector, Hu Moments.



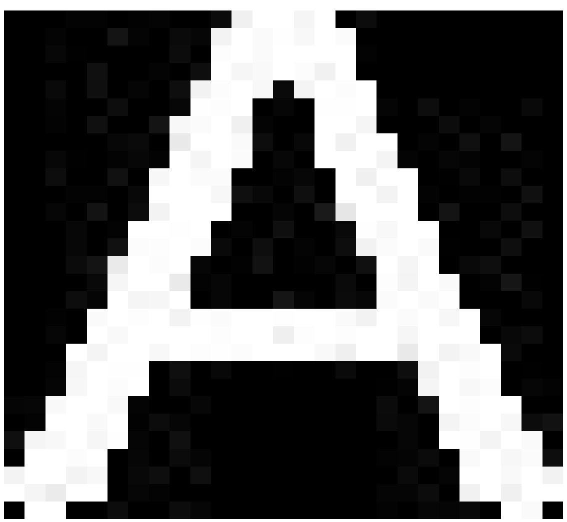
Feature : Mean X and Y Value

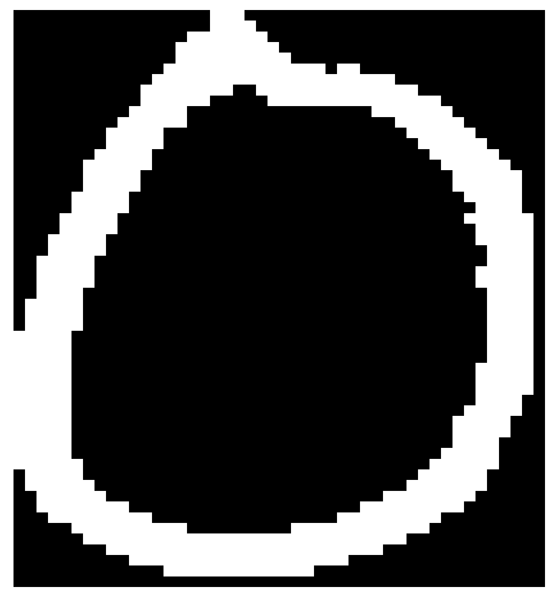
- For image f(x,y), mean X and mean Y value are defined as follows :

$$\bar{x} = \frac{\sum_x \sum_y f(x,y) \cdot x}{\sum_x \sum_y f(x,y)} \qquad \bar{y} = \frac{\sum_x \sum_y f(x,y) \cdot y}{\sum_x \sum_y f(x,y)}$$

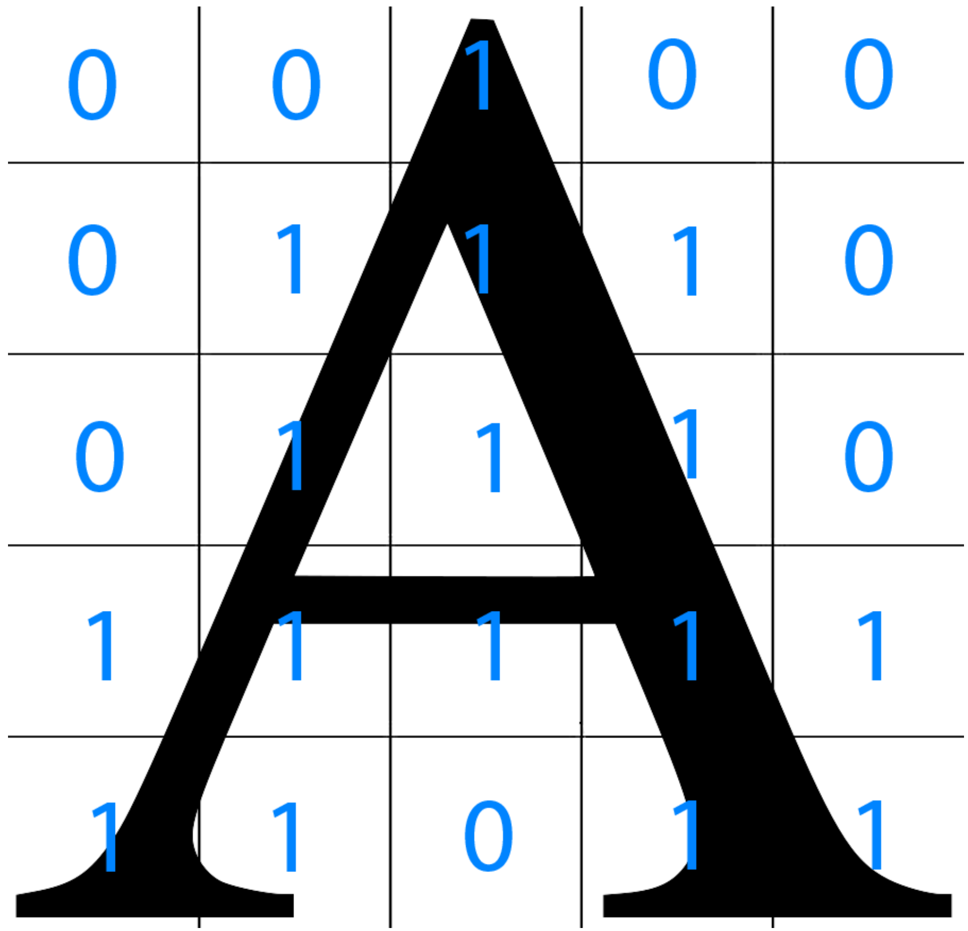
Feature : Number of Black Islands

- Example : A has 6 black islands, O has 3.





Feature : Grid Vector



Feature : Hu Moments

- Moment M of order (i+j) is defined as :

$$M_{ij} = \sum_{x=1}^{n} \sum_{y=1}^{n} x^i y^j I(x, y)$$

- This can be used to get Hu Moments which are invariant to translation, rotation and scaled. The 7 Hu Moments are :

$$H_1 = M_{20} + M_{02}$$
$$H_2 = (M_{20} - M_{02})^2 + 4(M_{11})^2$$
$$H_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2$$
$$H_4 = (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2$$
$$H_5 = (M_{30} - 3M_{12})(M_{30} + M_{12})$$
$$[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2]$$
$$+ 3(M_{21} - M_{03})(M_{21} + M_{03})$$
$$[3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2]$$

$$H_6 = (M_{20} - M_{02})$$
$$[(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2]$$
$$+ 4M_{11}(M_{30} + M_{12})(M_{21} + M_{03})$$

$$H_7 = (3M_{21} - M_{03})(M_{30} + M_{12})$$
$$[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2]$$
$$- (M_{30} - 3M_{12})(M_{21} + M_{03})$$
$$[3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2]$$

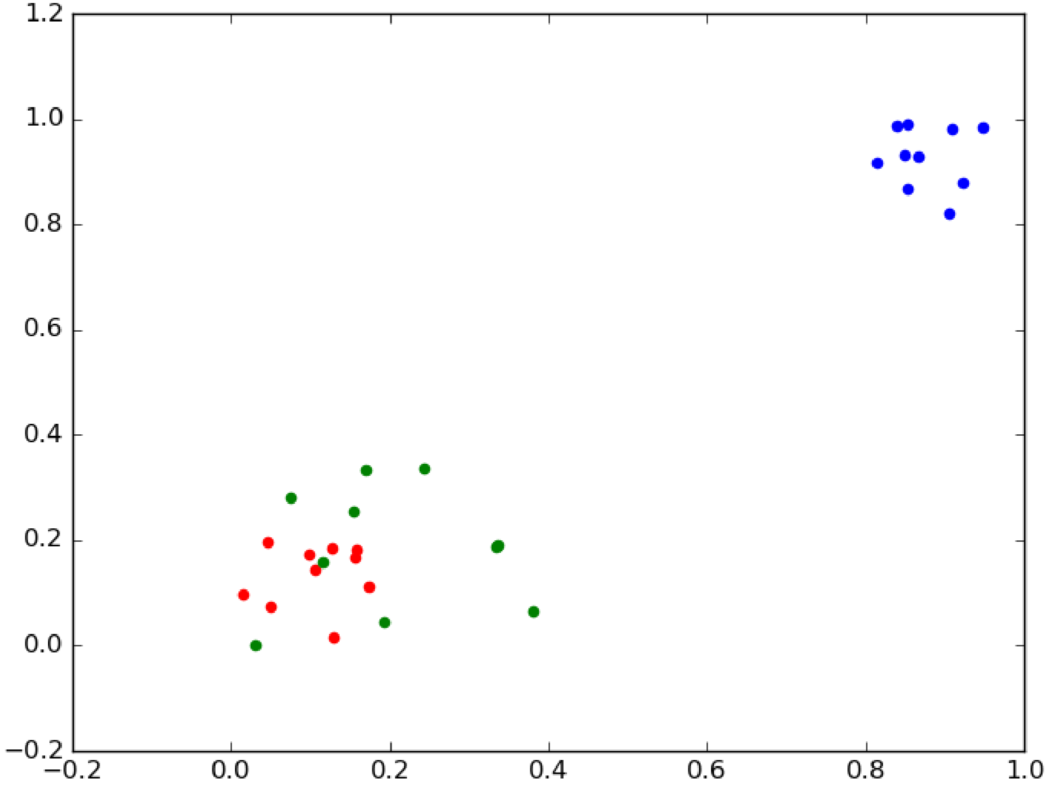
STEP 5 : Classification

- Features are supplied here to a classifier to get a class label, which indicates which character it possibly is.
- Methods used here are K-Nearest Neighbors Classifier and Support Vector Machines.
- For each model character, the features are extracted and labeled samples are used for training of the algorithm, which returns a model as output.

Classification : K-Nearest Neighbors

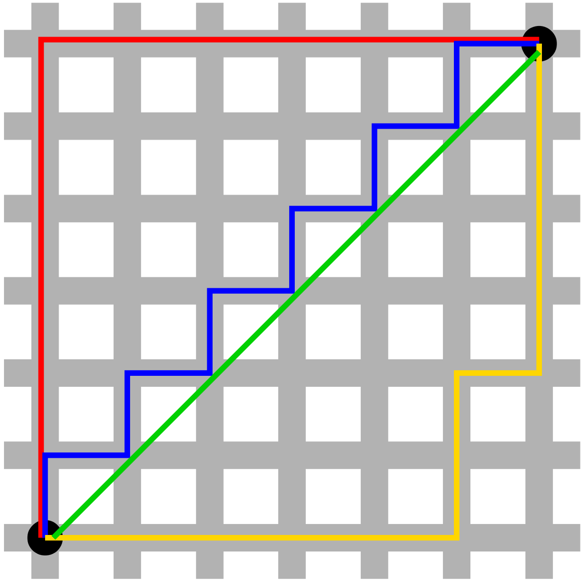
- It is a supervised learning algorithm.
- Each new instance gets assigned the class label which is most common among it's K Nearest Neighbors.
- It is a lazy learning algorithm, since KNN doesn't perform any operation on the training data till a query is received.
- Different distance measures : Taxi Cab Distance, Euclidean Distance, Minkowski Distance.

KNN Example



Distance Measure : Euclidean and Taxi Cab Distance

- Euclidean Distance : $\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}$
- Manhattan Distance : $|p_1 - q_1| + |p_2 - q_2| + \ldots + |p_n - q_n|$



Euclidean vs TaxiCab Distance [Wikipedia]

Distance Measure : Minkowski Distance

- Taxi Cab distance can also be written as

$$\left(|p_1 - q_1|^1 + |p_2 - q_2|^1 + ... + |p_n - q_n|^1\right)^{\frac{1}{1}}$$

  and Euclidean distance as

$$\left(|p_1 - q_1|^2 + |p_2 - q_2|^2 + ... + |p_n - q_n|^2\right)^{\frac{1}{2}}$$

- The general form, Minkowski Distance of order x is defined as follows :

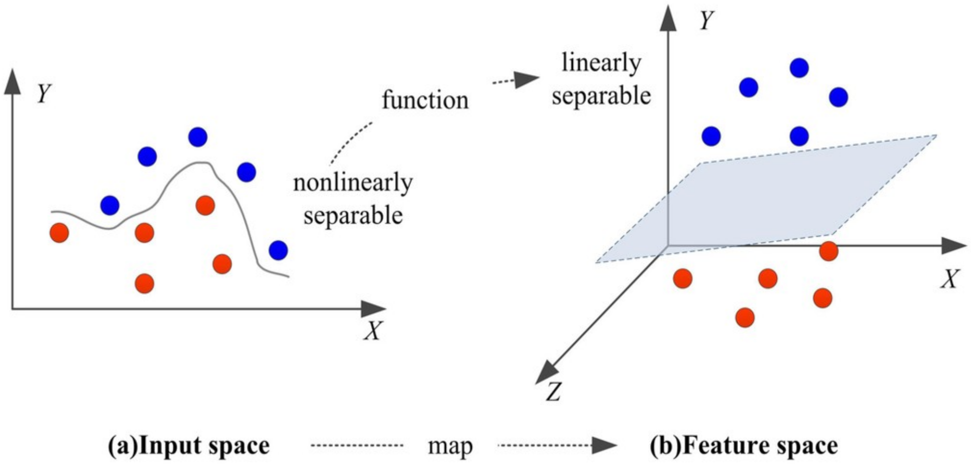
$$\left(|p_1 - q_1|^x + |p_2 - q_2|^x + ... + |p_n - q_n|^x\right)^{\frac{1}{x}}$$

Classification : Support Vector Machines

- It is a supervised learning algorithm.
- Constructs a set of hyperplanes which acts as decision boundaries.
- Boundaries used to decide which class an element belongs to.
- In its default form can be used for only binary classification problem.
- Multi-class classification is done by splitting the problem into multiple binary classification problems.

SVM : Mapping to Feature Space

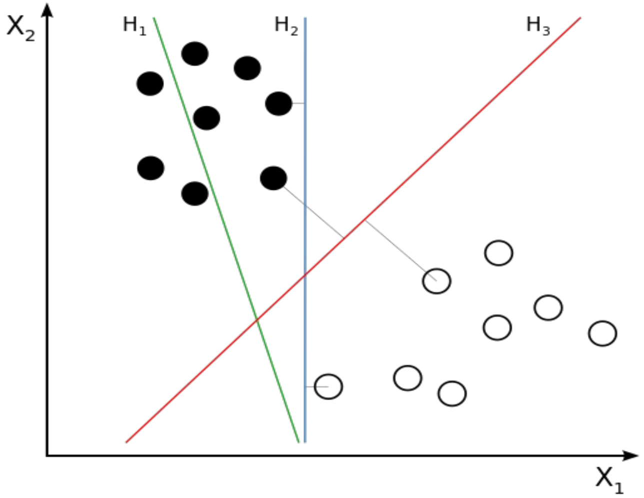
- In easy case, data is linearly separable, but this need not be always the case.
- In that scenario, SVM maps input to a higher dimensional feature space where the input is linearly separable.



Mapping Input space to Feature Space : [Cheng, Chun-Tian, et al]

SVM : Choosing Decision Plane

- There might be more than one plans which linearly separates the data.
- In such a case, SVM chooses the plane which maximizes the distance between training points and decision plane on either side.



Selecting Optimal Decision Plane : [Wikipedia]

STEP 6 : Post Processing

- Once words are detected, we check for them in a dictionary.
- If not present, could be case of wrong character detected.
- Suggest possible corrections based on similarity of characters.

Results : Computer Generated Text

TABLE I
Features Used vs Accuracy Attained for a 26 Class Classifier of Alphabets A-Z

Feature Used	Accuracy Attained
Mean X Value	0.45
Mean Y Value	0.36
Mean X and Y Value	0.72
Black Islands	0.09
Mean X and Y Value + Black Islands	0.81
Grid Vector	0.91
Hu Moments	0.54
All Combined	0.91

Results : Handwritten Text

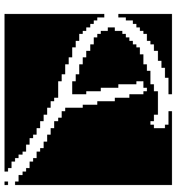
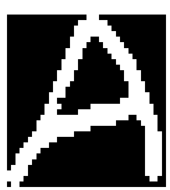
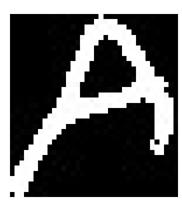
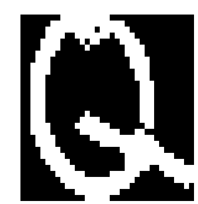
TABLE II

Features Used vs Accuracy Attained for a 26 Class Classifier of Alphabets A-Z

Features Used	Accuracy Attained
Mean X Value	0.2
Mean Y Value	0.4
Mean X and Y Value	0.4
Black Islands	0
Mean X and Y Value + Black Islands	0.2
Grid Vector	0.6
Hu Moments	0
All Combined	0.6

Summary

- The accuracy attained for computer generated images is higher than handwritten images in general.
- This makes sense since handwritten text does not have uniformity. Ratios not maintained, holes not maintained.

Summary

- The feature that works the best is the grid vector since it is not as sensitive to minor changes in the character.