BITS F464 - Machine Learning

Assignment 3

Naive Bayes Classifier

G V Sandeep	2014A7PS106H
Snehal Wadhwani	2014A7PS430H
Tanmaya Dabral	2014A7PS138H
Kushagra Agrawal	2014AAPS334H

BITS Pilani
Hyderabad Campus

Training of the Data

- The model first reads all the images (bitmaps) and stores them as a 1-D array of 4200 dimension (70X60).
- Then, on the basis of the labels (i.e. whether a human or not), the program updates a hash table as follows:
  - Hash Table contains the following **key,value** pair:
    key - The pixel (array position) being considered
    value - A 1-D (`Arr`) array with 4 elements.
  - `Arr[0]` : stores the count of number of instances where the pixel at that position was blank (0) and the bitmap was not classified as a human (0).
  - `Arr[1]` : stores the count of number of instances where the pixel at that position was blank (0) and the bitmap was classified as a human (1).
  - `Arr[2]` : stores the count of number of instances where the pixel at that position was **'#'** (1) and the bitmap was not classified as a human (0).
  - `Arr[3]` : stores the count of number of instances where the pixel at that position was **'#'** (1) and the bitmap was classified as a human (1).
- Once the entire hash table has been populated with these values, the program converts these counts to corresponding probabilities in the following manner :

```
Arr[0] = (Arr[0])/(Arr[0]+Arr[2])
Arr[1] = (Arr[1])/(Arr[1]+Arr[3])
Arr[2] = (Arr[2])/(Arr[0]+Arr[2])
Arr[3] = (Arr[3])/(Arr[1]+Arr[3])
```

- Now these elements have the following meaning :
  - `Arr[0]` : The probability of the pixel being blank(0) given the image is not classified as a human.
  - `Arr[1]` : The probability of the pixel being blank(0) given the image is classified as a human.
  - `Arr[2]` : The probability of the pixel being **'#'** (1) given the image is not classified as a human.
  - `Arr[3]` : The probability of the pixel being **'#'** (1) given the image is classified as a human.

Prediction

- Given a testing instance of a bitmap the program runs through the sequence of pixels and calculates the probability of being classified as a human and that of not being classified as human. The program then gives the label having higher probability as its prediction. This probability is calculated using Bayes Theorem :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

In this case, A is the output label 'human' or 'not human'; B is the the list of features (i.e. value of each pixel). The list of features in B are considered to be independent and the compound probability is computed by multiplication of the individual feature having a particular value given the output label.

- Note that to avoid working with very small numbers and facing difficulties because of rounding off errors, the program rather calculates the log of probabilities and takes their sum instead of direct multiplication.

- Apart from the classical Naive Bayes Algorithm described above, *Additive smoothing* or *Laplace Smoothing* has been applied which is given by the following formula :

$$P(feature|label) = \frac{x + \alpha}{N + \alpha d} \tag{2}$$

Where x is the no of instances satisfying the feature given the output label and N is the no. of instances with the given output label. α (smoothing constant) is taken as 1 and d is this problem is 2. When α is 1, this smoothing is also called *add-one smoothing*.

Results

Without Smoothing

Accuracy: 88.67 %

Confusion Matrix :

	True	**False**
Positive	64	8
Negative	69	9

With Add-one Smoothing

Accuracy: 90.0 %

Confusion Matrix :

	True	**False**
Positive	68	10
Negative	67	5

Examples of False Negatives and False Positives

The following two images were classified as *not human* (i.e. they are False Negatives) by the Naive Bayes Classifier but were rightly caught as being *human* upon smoothing.

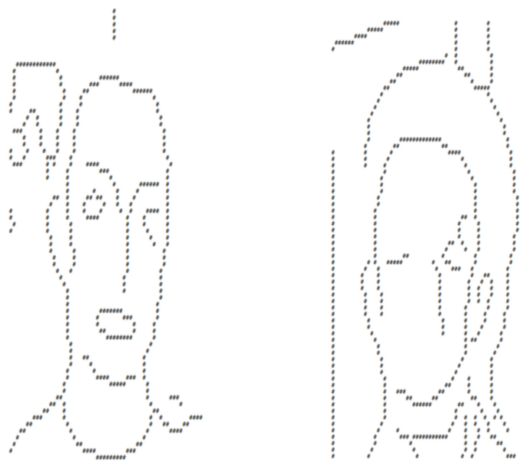

Figure 1: False Negatives by Naive Bayes without smoothing

The following two images were classified as *human* (i.e. They are False Positives) by the Naive Bayes Classifier with smoothing but were rightly classified as *not human* by the one without smoothing.

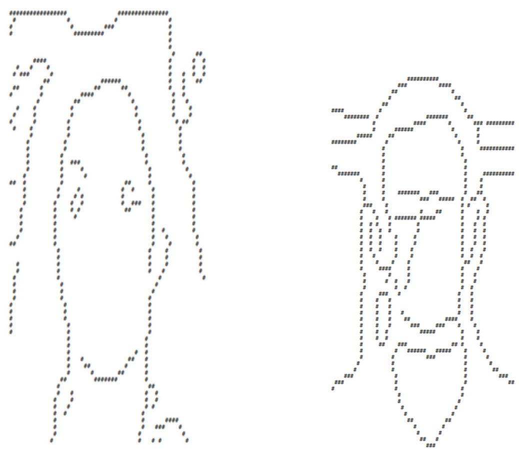

Figure 2: False Positives by Naive Bayes with smoothing

These observations clearly show that since smoothing even attempts to classify the remotest possibilities of human-like figures as *humans*, it drastically improves the quality of the categorisation. Also, the normal Naive Bayes classifier has missed out on a few, clear human-looking figures due to little deformities which, again, have been correctly classified by the Naive Bayes classifier that implements smoothing.