Cairo University
Faculty of Engineering.
Credits Hours System

CMPN303
Fall 2020

Pattern Recognition and Neural Networks Project

Writer Identification System

Done by:

Ahmed Ibrahim Abdel Latif	1170024
Ahmed Osama Mohamed	1170504
Moaaz Ashraf Zaki	1170353
Mohammed Mohammed Saad	1170044

I. PIPELINE

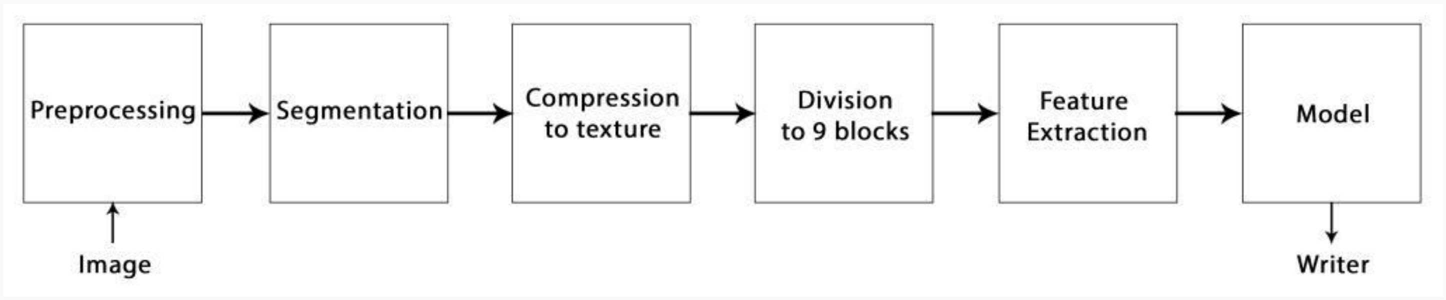


Figure 1.1

1-Preprocessing:

Extracts only the written text from the image and gets rid of any typed text and white space.

2-Segmentation:

Gets all the connected components in the image and the y-coordinate line boundaries of each line.

3-Compression to texture:

Coverts the written text image into a texture image.

4-Division to 9 blocks:

Divides the texture image into 9 equally sized texture blocks.

5-Feature Extraction:

Obtains the LBP features for each texture block.

6-Model:

Trains on the obtained features and predicts the results.

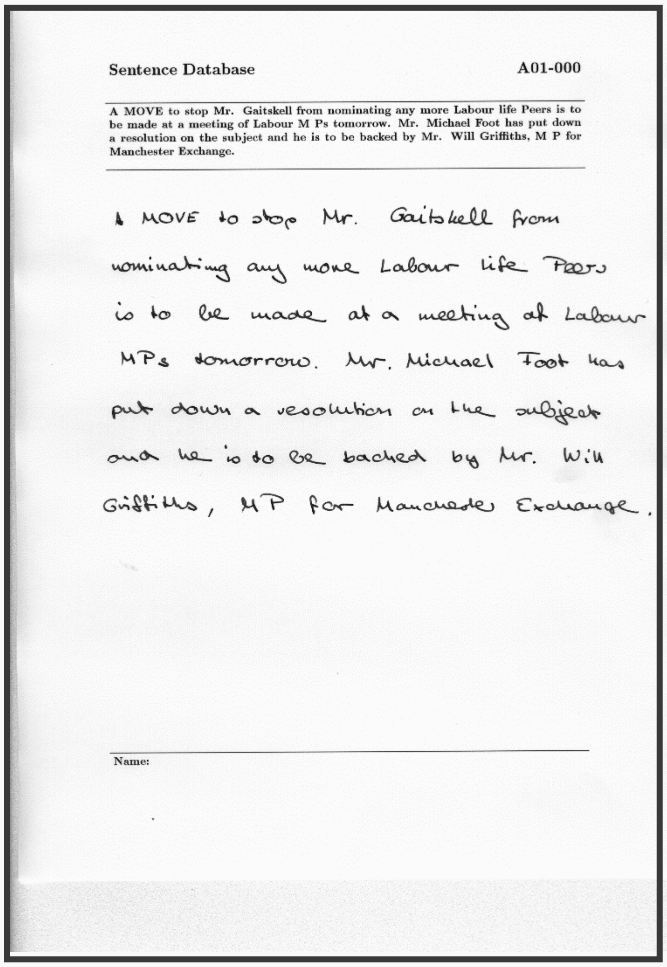
II. PREPROCESSING

Preprocessing is one of the most important modules in our pipeline, because it ensures that the image that we are working on follows a certain structure that is mendatory for the other modules to work properly.

It takes no detective to notice the repeated pattern in all the IAM dataset images. A page header. Typed text that is bounded between two horizontal lines with roughly the same width. The written text. Another horizontal line. A footer. There is white gap at the left side that is fixed in all papers. And on the right side, the white gap width changes a little from one paper to another.

Our objective was clear. We needed a way to detect the horizontal lines y-coordinates so we can get rid of the typed text and it was done by detecting the contours of the whole image, filtering their bounding boxes based on the width and height, and then chosing the most lower bounding rectancle in the upper half of the image and the most uppper one in the lower part. Now our image contains only the written text. But we still needed to get rid of the white space surrounding it. Since the written text was the only black pixels left, a simple horizontal and vertical histogram were enough to help us do just that after erroding the image to git rid of any background noise first.

<u>Original sample: Preprocessed sample:</u>


Sentence Database A01-000

A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour M Ps tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, M P for Manchester Exchange.

A MOVE to stop Mr. Gaitskell from
nominating any more Labour life Peers
is to be made at a meeting of Labour
MPs tomorrow. Mr. Michael Foot has
put down a resolution on the subject
and he is to be backed by Mr. Will
Griffiths, MP for Manchester Exchange.

Name:

Figure 2.2 original image

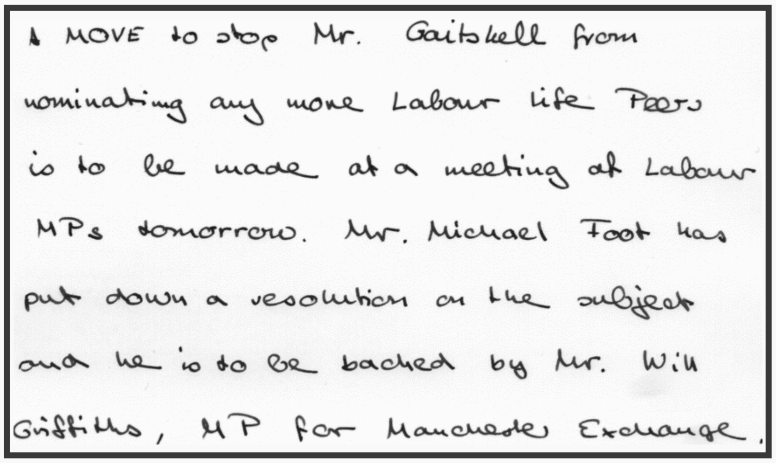

A MOVE to stop Mr. Gaitskell from
nominating any more Labour life Peers
is to be made at a meeting of Labour
MPs tomorrow. Mr. Michael Foot has
put down a resolution on the subject
and he is to be backed by Mr. Will
Griffiths, MP for Manchester Exchange.

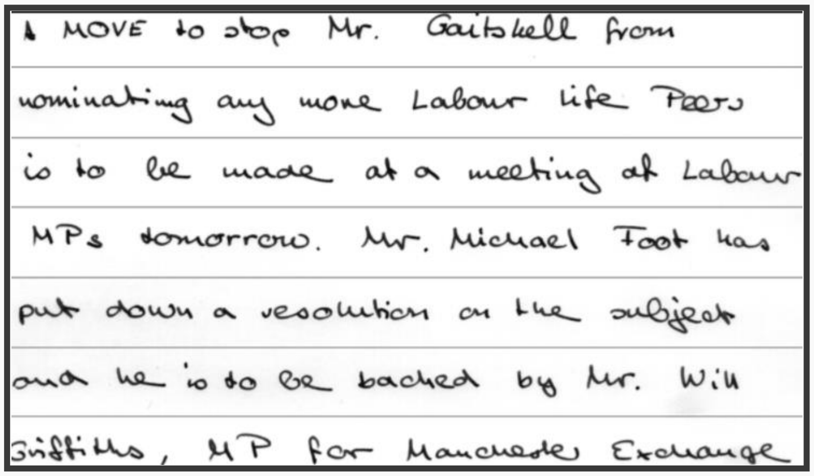
Figure 2.2 preprocessed image

III. SEGMENTATION, COMPRESSION & DIVISION

Our approach to the problem essentially turns the image into a texture image. Then divides this texture image into equally sized texture blocks.

We do such a thing by first getting the connected components of the image, filter these components based on a certain threshold to get rid of any commas or periods. We then scan the image line by line and move the connected components in that line to a new image. And each time the y coordinate of the new line in the new image is calculated according to this equation. $y = y_{prev} + \frac{h}{2}$ where, y_{prev} is the y coordinate of the previous line and h is the average height of the connected components in the previous line.

<u>Getting the separating lines to be able to move line by line:</u>



A MOVE to stop Mr. Gaitskell from
nominating any more Labour life Peers
is to be made at a meeting of Labour
MPs tomorrow. Mr. Michael Foot has
put down a resolution on the subject
and he is to be backed by Mr. Will
Griffiths, MP for Manchester Exchange

Figure 3.1 separating lines

<u>The obtained texture image:</u> ***<u>The obtained texture blocks:</u>***

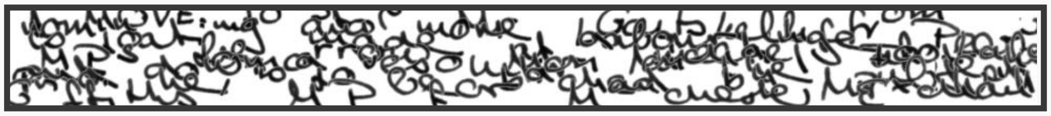


Figure 3.2 texture image

After creating the texture image, we then divide this image into 9 equal blocks as illustrated in figure 3.3. These generated blocks of texture are considered the new input from which we will obtain our LBP features in the next section.

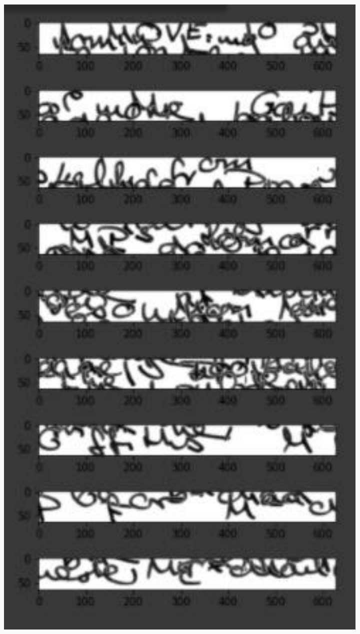


Figure 3.3 texture blocks

IV. FEATURE EXTRACTION

First Approach (LBP):

It’s no secret that the LBP features are powerful when it comes to classifying between different textures, and so far, all what we have been doing is transforming our written text image into texture blocks to make use of this fact. The way the LBP features are extracted from the image is essentially looking at each pixel and comparing it to its neighboring pixels, if its value is greater than that of one of its neighbors, this yields zero to that corresponding neighbor, if not, it yields one. And the same is done for all the other neighbors. As illustrated in figure 4.1. We then take the summation of each binary value multiplied by two to the power of that neighbor index relative to the center pixel as illustrated in figure 4.2.

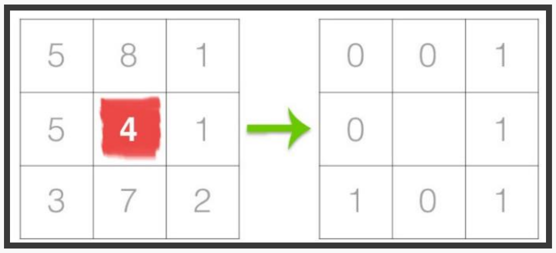


Figure 4.1 binary representation of LBP

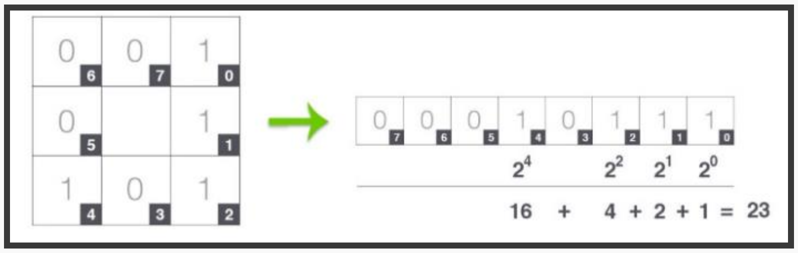


Figure 4.2 LBP center pixel value

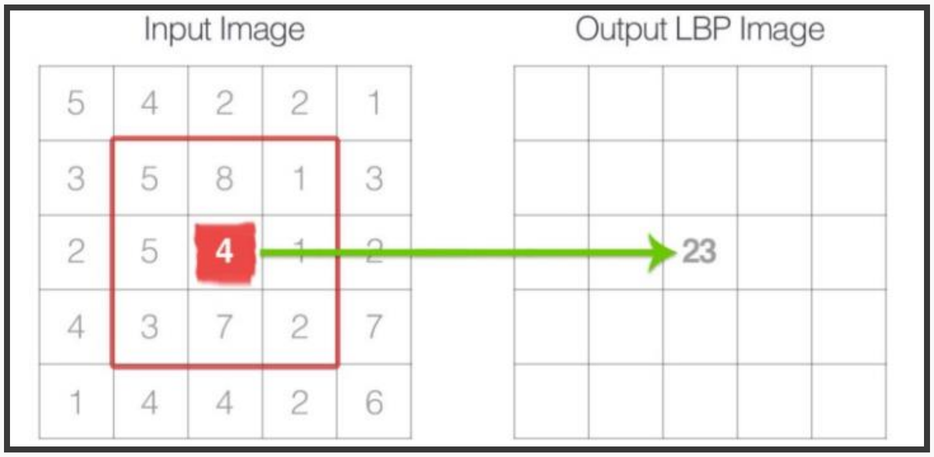


Figure 4.3 LBP produced Image

From figure 4.3, we can obtain a histogram out of the values developed in the LBP image as they all range from 0 to 255, we count the frequency of each number to produce a 256 bins histogram which now represents our 256 feature vector.

Now there are two parameters to consider in the LBP:

1. The number of points in a circularly symmetric neighborhood.
2. The radius of the circle to account for different scales.

Figure 4.4 illustrates how chosing different values for the radius and number of points may change what we have discussed above.

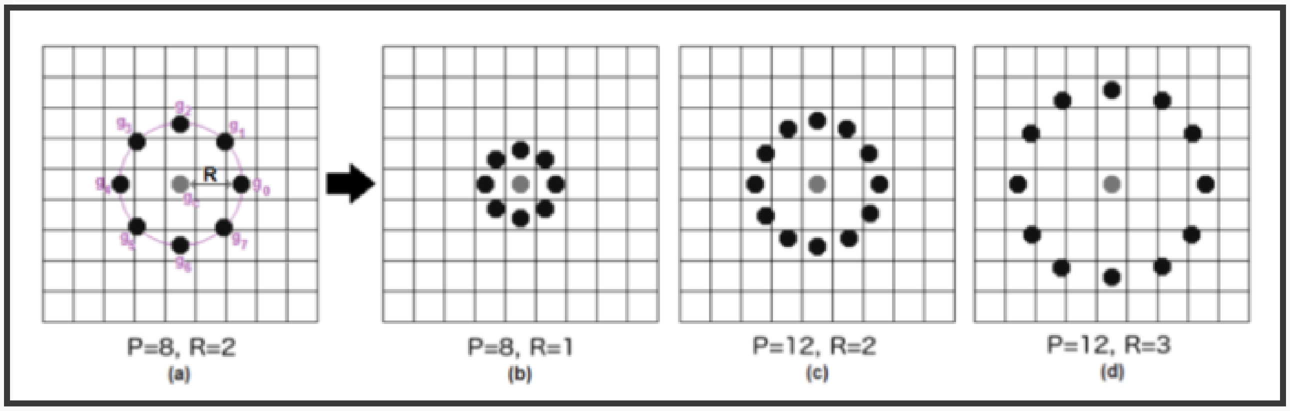


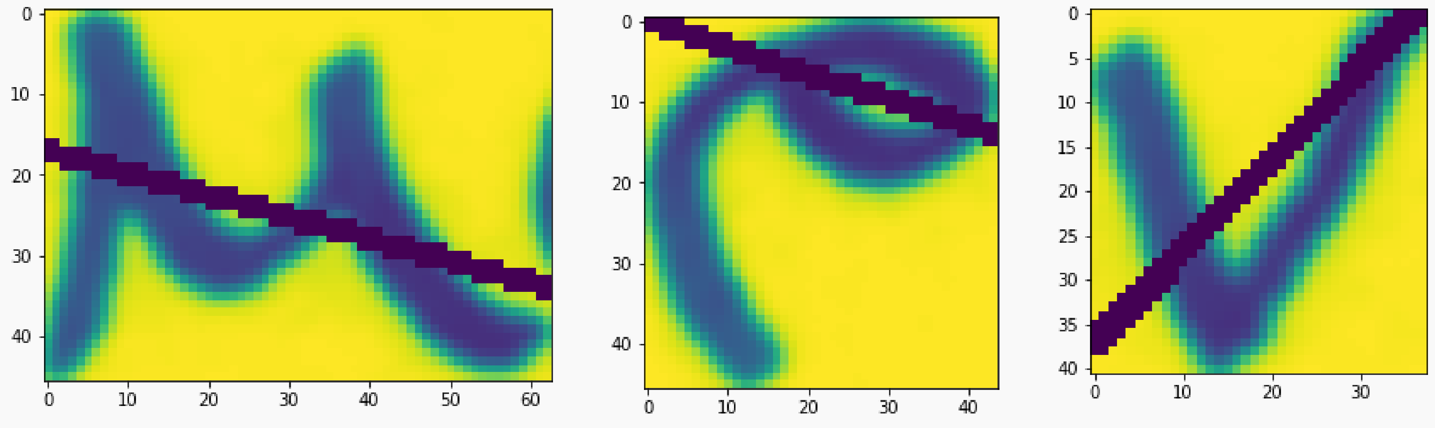
Figure 4.4 radius and number of points representations

In our approach, we have stuck to the traditional number of points taken which is 8, but we have tried out different values for the radius in our validation set to account for the different possible scales, and found out that a radius of 5 best represents our developed scale. The exact accuracy results obtained from trying out different radii is illustred in details in the performance analysis section.

Second Approach (Connected Components Properties):

Starting from having the connected components and its properties. The statistics about the connected components consists of upper value of the bounding box, left value, width and height. Summing and averaging each of the width values and height values, gave us two features representing the average heights and average widths of the bounded boxes of the connected components.

Another thinking was about calculating the orientation of the letters or words written within each component. Using the contours extracted from each connected component, we could fit a line with an angle representing the orientation or inclination of what is written inside the component. We used the info we have about the line to get the angle which is our third feature.



V. MODEL SELECTION

Table 5.1

Model	Iterations	Accuracy
KNN (k=5)	**10**	**100%**
KNN (k=5)	**100**	**98%**
SVM(C=8)	**10**	**100%**
SVM(C=8)	**100**	**100%**
SVM(C=8)	**1000**	**98.7%**
Random Forest (max depth)	**10**	**100%**
Random Forest (max depth)	**100**	**96%**
Gradient Boosting (learning rate=0.05, n estimators=1000, max depth=32)	**10**	**100%**
Gradient Boosting (learning rate=0.05, n estimators=1000, max depth=32)	**100**	**99%**
Gradient Boosting (learning rate=0.05, n estimators=1000, max depth=32)	**1000**	**95.8%**

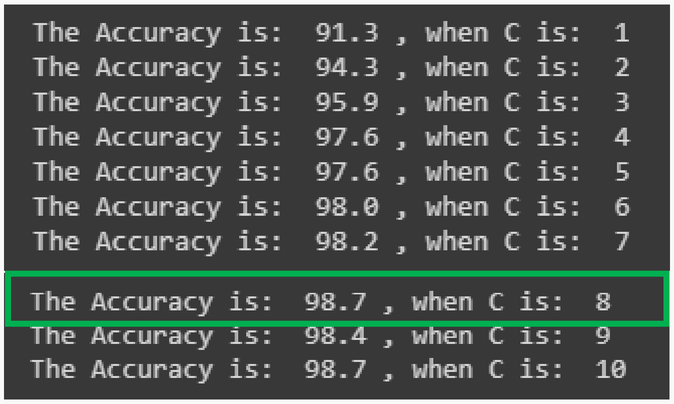
From table 5.1, we can clearly see that the SVM model out performed all the other ones, and based on this fact we have decided to use it.

VI. PERFORMANCE ANALYSIS

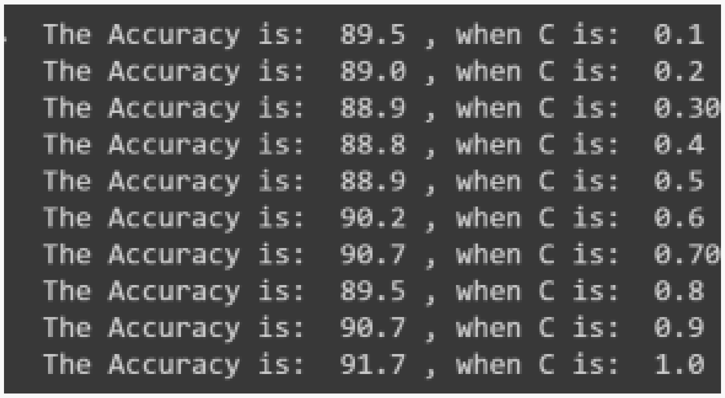
Performance Results:

<u>Trying out different Cs for SVM with 1000 iterations</u>

<u>from range 1→10 and step 1 from range 0.1→1 and step 0.1</u>

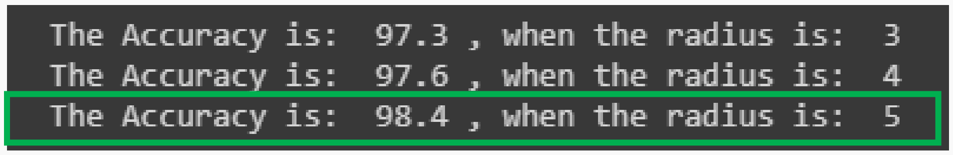


```
The Accuracy is:  91.3 , when C is:  1
The Accuracy is:  94.3 , when C is:  2
The Accuracy is:  95.9 , when C is:  3
The Accuracy is:  97.6 , when C is:  4
The Accuracy is:  97.6 , when C is:  5
The Accuracy is:  98.0 , when C is:  6
The Accuracy is:  98.2 , when C is:  7
The Accuracy is:  98.7 , when C is:  8
The Accuracy is:  98.4 , when C is:  9
The Accuracy is:  98.7 , when C is:  10
```



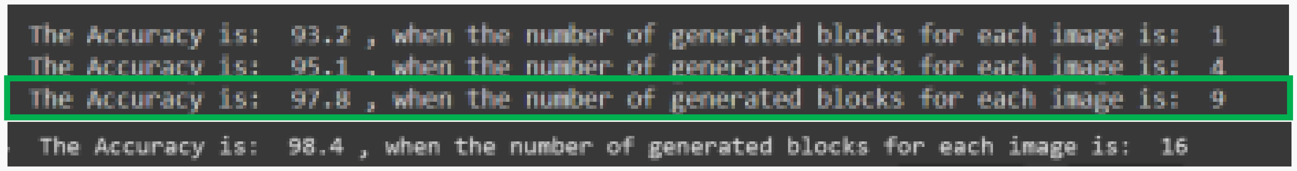
```
The Accuracy is:  89.5 , when C is:  0.1
The Accuracy is:  89.0 , when C is:  0.2
The Accuracy is:  88.9 , when C is:  0.30
The Accuracy is:  88.8 , when C is:  0.4
The Accuracy is:  88.9 , when C is:  0.5
The Accuracy is:  90.2 , when C is:  0.6
The Accuracy is:  90.7 , when C is:  0.70
The Accuracy is:  89.5 , when C is:  0.8
The Accuracy is:  90.7 , when C is:  0.9
The Accuracy is:  91.7 , when C is:  1.0
```

<u>Trying out different radii for the LBP [3,4,5] with 1000 iterations</u>



```
The Accuracy is:  97.3 , when the radius is:  3
The Accuracy is:  97.6 , when the radius is:  4
The Accuracy is:  98.4 , when the radius is:  5
```

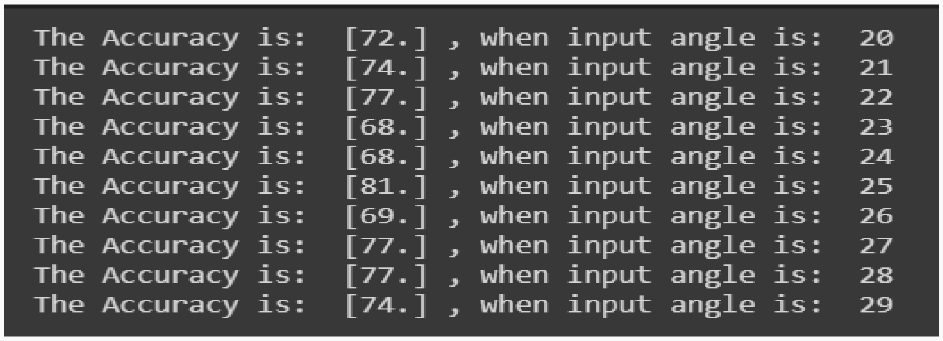
<u>Trying out different divisions for the segmentation [1,2,3,4] with 1000 iterations</u>



```
The Accuracy is:  93.2 , when the number of generated blocks for each image is:  1
The Accuracy is:  95.1 , when the number of generated blocks for each image is:  4
The Accuracy is:  97.8 , when the number of generated blocks for each image is:  9
The Accuracy is:  98.4 , when the number of generated blocks for each image is:  16
```

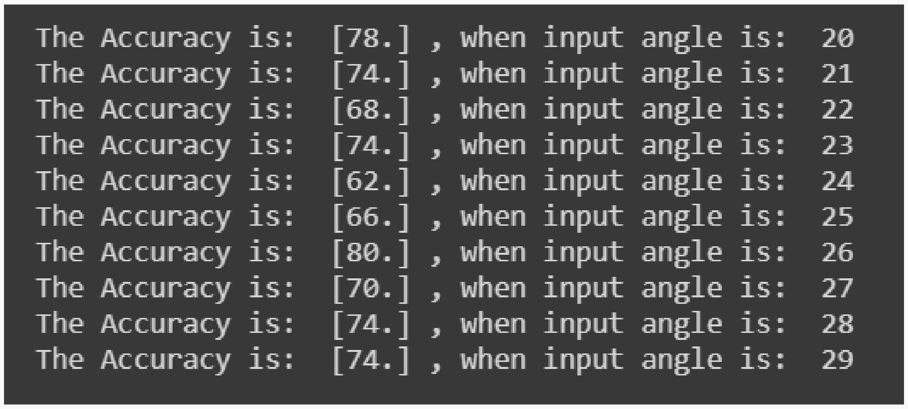
Second Approach (Connected Components Properties):

<u>Trying out different input angles (threshold) with constant C=5.0 for SVM with 100 iterations</u>



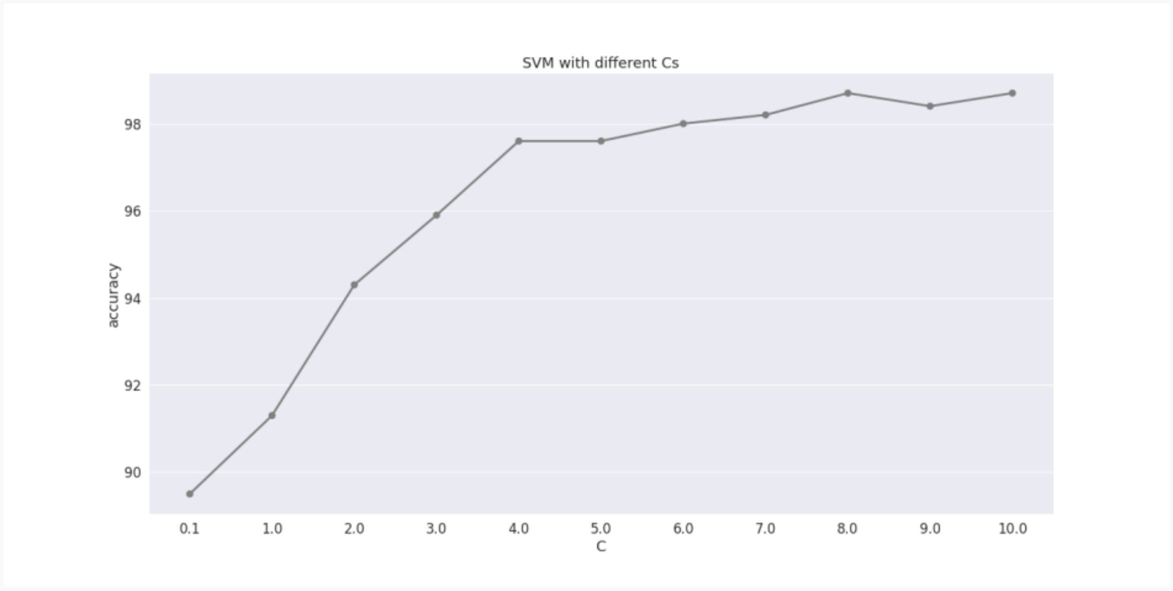
```
The Accuracy is:  [72.] , when input angle is:  20
The Accuracy is:  [74.] , when input angle is:  21
The Accuracy is:  [77.] , when input angle is:  22
The Accuracy is:  [68.] , when input angle is:  23
The Accuracy is:  [68.] , when input angle is:  24
The Accuracy is:  [81.] , when input angle is:  25
The Accuracy is:  [69.] , when input angle is:  26
The Accuracy is:  [77.] , when input angle is:  27
The Accuracy is:  [77.] , when input angle is:  28
The Accuracy is:  [74.] , when input angle is:  29
```

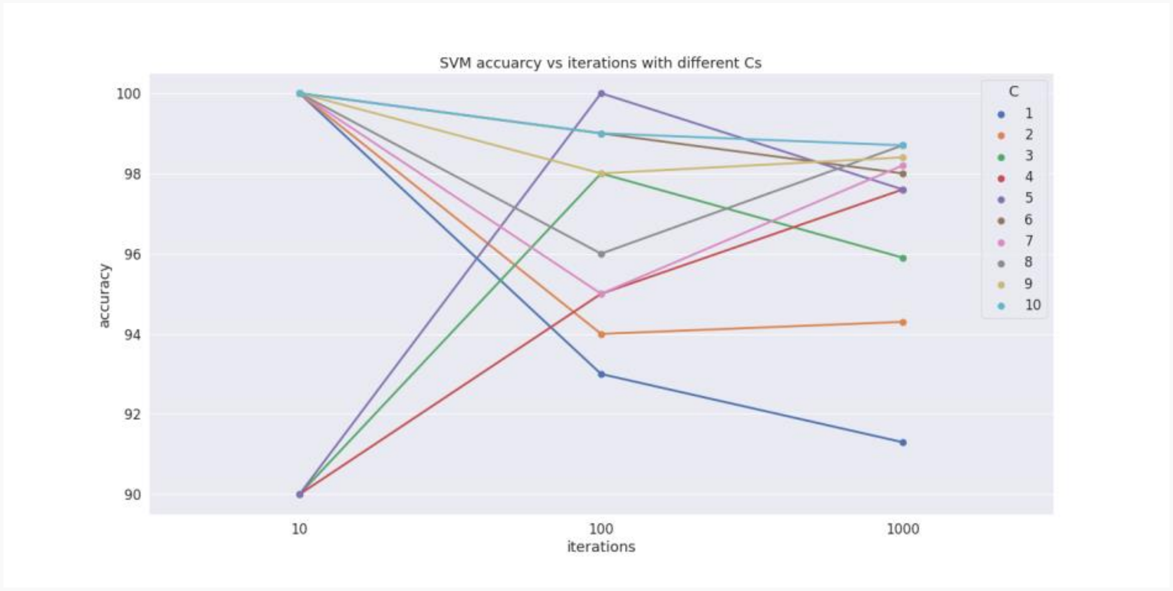
<u>Trying out different input angles (threshold) with constant C=5.0 for SVM with 50 iterations</u>



```
The Accuracy is:  [78.] , when input angle is:  20
The Accuracy is:  [74.] , when input angle is:  21
The Accuracy is:  [68.] , when input angle is:  22
The Accuracy is:  [74.] , when input angle is:  23
The Accuracy is:  [62.] , when input angle is:  24
The Accuracy is:  [66.] , when input angle is:  25
The Accuracy is:  [80.] , when input angle is:  26
The Accuracy is:  [70.] , when input angle is:  27
The Accuracy is:  [74.] , when input angle is:  28
The Accuracy is:  [74.] , when input angle is:  29
```

Visualizations:





VII. ENHANCEMENTS AND FUTURE WORK

Since our LBP features mainly depend on differentiating between textures. The model may confuse any two authors who would have a similar texture after compressing the image. Though this might be a rare case, but this could have been enhanced by adding more features beside the LBP ones that would focus more on this weakness.

VIII. PREVIOUS WORK

- **Preprocessing:**
  - After converting the image to grayscale (figure #1) , a separation from the background was conducted using a threshold value returned by the threshold_otsu() function provided by the skimage library (figure #2), afterwards we got the edges using the canny() function provided also by the skimage library (figure #3)

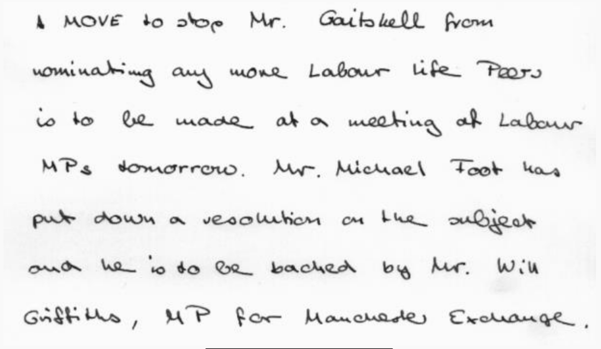

A MOVE to stop Mr. Gaitskell from
nominating any more Labour life Peers
is to be made at a meeting of Labour
MPs tomorrow. Mr. Michael Foot has
put down a resolution on the subject
and he is to be backed by Mr. Will
Griffiths, MP for Manchester Exchange.

Figure #1

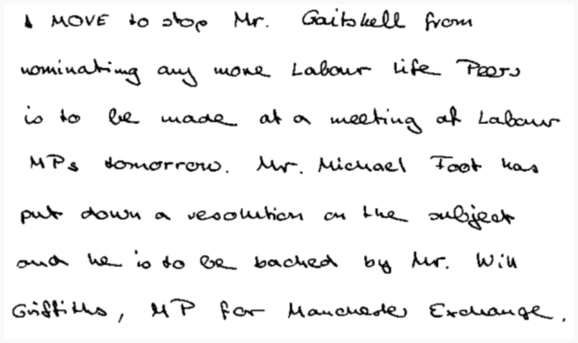

A MOVE to stop Mr. Gaitskell from
nominating any more Labour life Peers
is to be made at a meeting of Labour
MPs tomorrow. Mr. Michael Foot has
put down a resolution on the subject
and he is to be backed by Mr. Will
Griffiths, MP for Manchester Exchange.

Figure #2

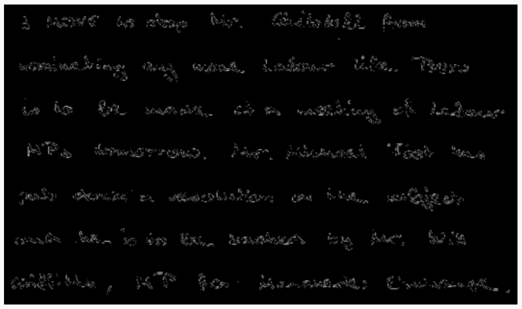
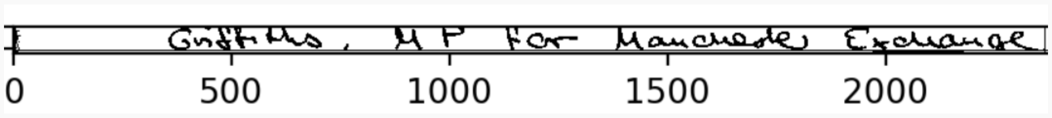

A MOVE to stop Mr. Gaitskell from
nominating any more Labour life Peers
is to be made at a meeting of Labour
MPs tomorrow. Mr. Michael Foot has
put down a resolution on the subject
and he is to be backed by Mr. Will
Griffiths, MP for Manchester Exchange.

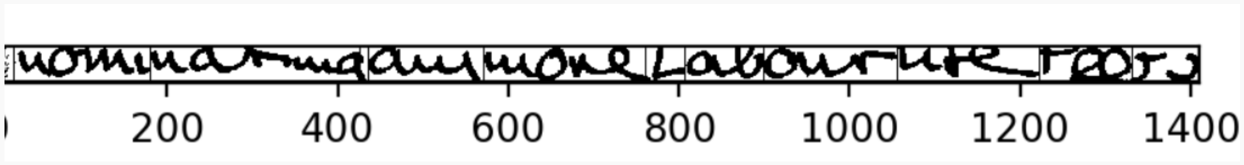
Figure #3

- **Getting the rows containing handwritten text (segmentation)**
  - Now that we have the matrix containing the edges of the letters, we used it in getting the histogram representing the pixels of written text per rows in the has helped us filter the rows of the grayscale image (separated from background) and keep the ones of a histogram value greater than a certain value, then stacking a row of black pixels between each line to simulate a boundary between lines.
  - Now that we have the index of each boundary, we can slice using the index of each two consecutive boundaries and get the line in between, we can apply feature extraction accordingly

- The following figure shows an example of the result we got from this stage



- **Feature Extraction**
  - The first feature we extracted is the height of each line, since we have stacked a line of black pixels the lines, we can now get the indices of the rows representing the boundaries and get the difference between the indices of the consecutive ones to represent the height. We store the height of each line in an array, thus at the end we can get the average height for this writer
  - The next feature we extracted was the width of the connected components (or the letters written as a whole shape with no space between them) and this was done through the following stages
    - **Segmenting each line into connected components**
      - This is done using a histogram representing the pixels of letters in each column, which allowed us to filter the columns and keep the ones of a value greater than a threshold. If consecutive columns are kept, this means that these columns together represent a shape that is written as a connected component that may be a group of letters written with no spaces between them.
      - We stacked columns of black pixels between the connected components to represent the boundaries between them. The following figure shows the result of this process

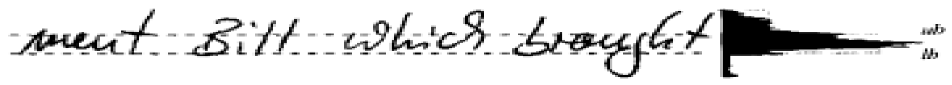


      - Now that we have the indices of these columns representing the boundaries of the connected components, we are able to calculate the width of each connected component
      - We stored the width of all the connected components in an array, thus we can calculate the average width of connected components of this writer

  - We wanted to try applying some idea from one of the papers about histogram of the writing zones and also features about height of writing areas.

- **Paper citation:** Marti, Urs-Viktor & Messerli, R. & Bunke, H.. (2001). Writer identification using text line based features. Proc. ICDAR'01. 101 - 105. 10.1109/ICDAR.2001.953763.

Figure 1. Writing zones and bounding lines.



IX. WORKLOAD DISTRIBUTION

We divided ourselves to two teams to work in two separate approaches in parallel, and took the best results at the end.

Team members	Task
Moaaz Zaki & Mohammed Saad	**Preprocessing – Connected components – Lines detection– Texture blocks – LBP – Model selection**
Ahmed Ibrahim & Ahmed Osama	**Line detection (another trial) – Width of the component – Height of the component – Slant (value and direction / above threshold)**

X. REFERENCES

- *Writer identification using texture features: A comparative study by Priyanka Singh, Partha Pratim Roy, Balasubramanian Raman.*
- *Writer verification using texture-based features R. K. Hanusiak · L. S. Oliveira · E. Justino R. Sabourin.*
- *https://medium.com/@ariesiitr/texture-analysis-using-lbp-e61e87a9056d*