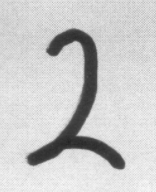
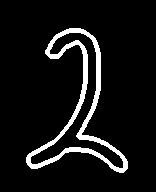
**EE4266 Assignment 3 Report**

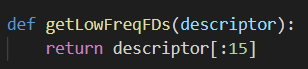
Name: Tan Fu Long U1820505D

Teammate Name: Tan Wei Hao

**Obtaining contours**

1. The letter “C” and “2” were cropped out and saved as image files to be used as templates.  
   
2. The image files for “C”, “2” and sample were converted to grayscale and then inverted.
3. Binary thresholding is applied to convert the grayscale image with varying luminosities to a binary image in which pixels with luminosity greater than the threshold are converted to white and the other pixels to black. Steps 2 and 3 aid the edge detection algorithm and contour detection in openCV.
4. The images were dilated to increase the area of the object and accentuate its features.  
   
5. The contours for the three images were obtained by calling the function, cv2.findContours().  
   
6. The contour vectors (x, y) were obtained by iterating through the contour and appending the vector of each point in an array. The elements in the array were then converted to complex numbers in the form of x+jy.

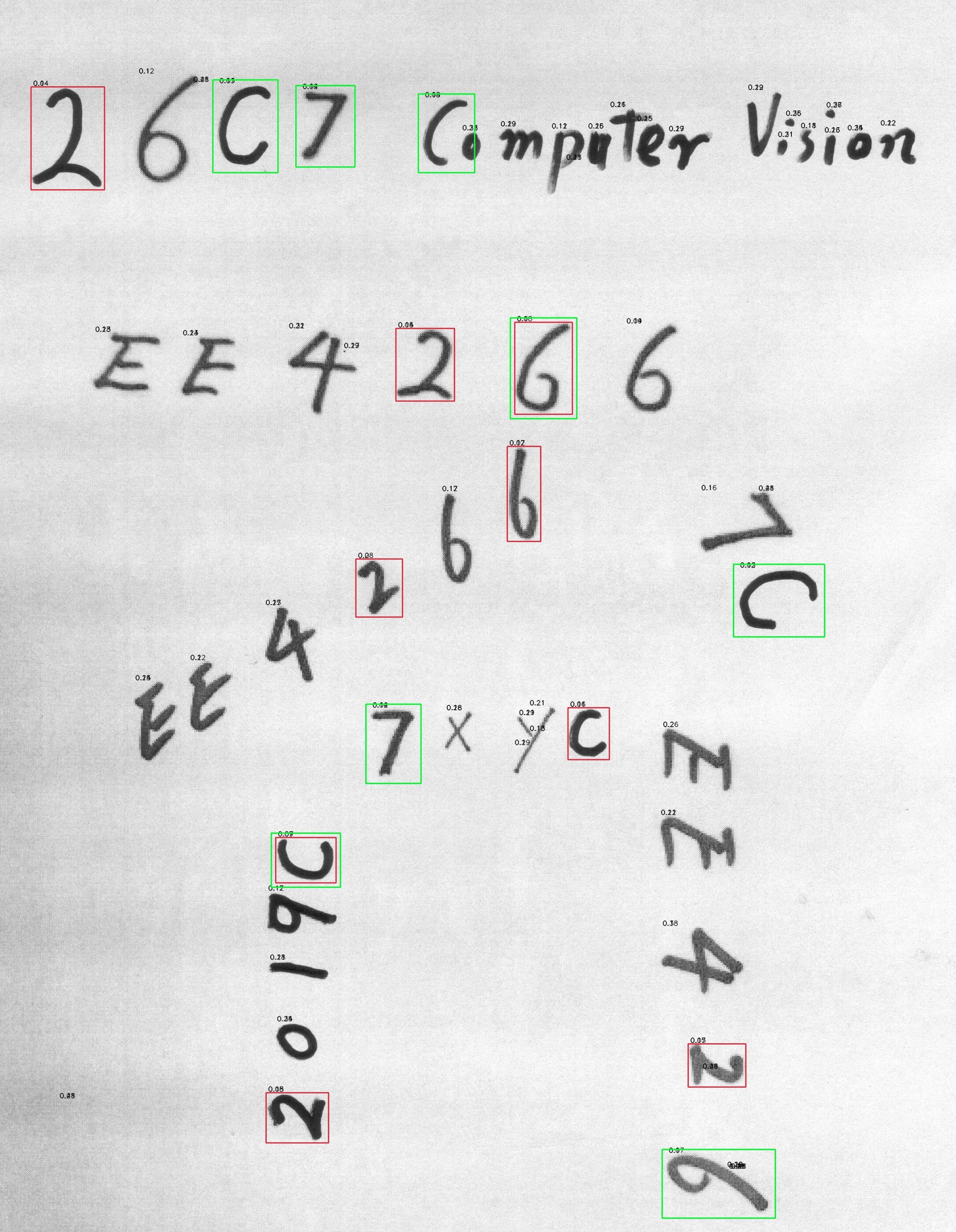
**Discrete Fourier Transform**

1. The Discrete Fourier Transform of the complex contour vectors were evaluated.
2. To make the Fourier Descriptors invariant to the following factors, the steps below were taken:
   1. Rotation - Convert the Fourier Descriptor from a+jb form to r∠𝜃 (magnitude phase) form. Extract only the magnitude component since phase is not considered in subsequent calculations.  
      
   2. Scale - Divide all Fourier Descriptors by the magnitude of the first descriptor to normalize the size of the contour  
      
   3. Translation - For closed contours, the sum of the sinusoidal terms is zero. The shifting of the origin to the center of the contour was achieved by excluding the dc (zero) term.  
      
   4. Initial point - Proven to be invariant to initial point. No modifications necessary.
3. The high frequency components were removed to decrease computational complexity.
4. The inverse FFT of the remaining lower frequency components were evaluated and drawn to ensure that sufficient detail still remained.  
   

**Thresholding**

1. The Euclidean distance between the Fourier Descriptors of the sample and templates were calculated. The descriptors in the sample that were below the threshold would be considered as a match for the character.
2. The threshold values for the characters are as below:
   1. “2” - 0.09
   2. “C” - 0.1

**Preliminary Results**



1. Overall and individual results are broken down as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Result | Accuracy (%) | Precision (%) | Recall (%) |
| ‘2’ | 91.3 | 55.6 | 100 |
| ‘C’ | 89.1 | 50.0 | 80.0 |
| Overall | 88.9 | 64.3 | 90.0 |

1. Although the algorithm correctly identified multiple ‘2’ and ‘C’s, there were many false positives originating from the misdetection of numbers 6 and 7. This explains the low precision.

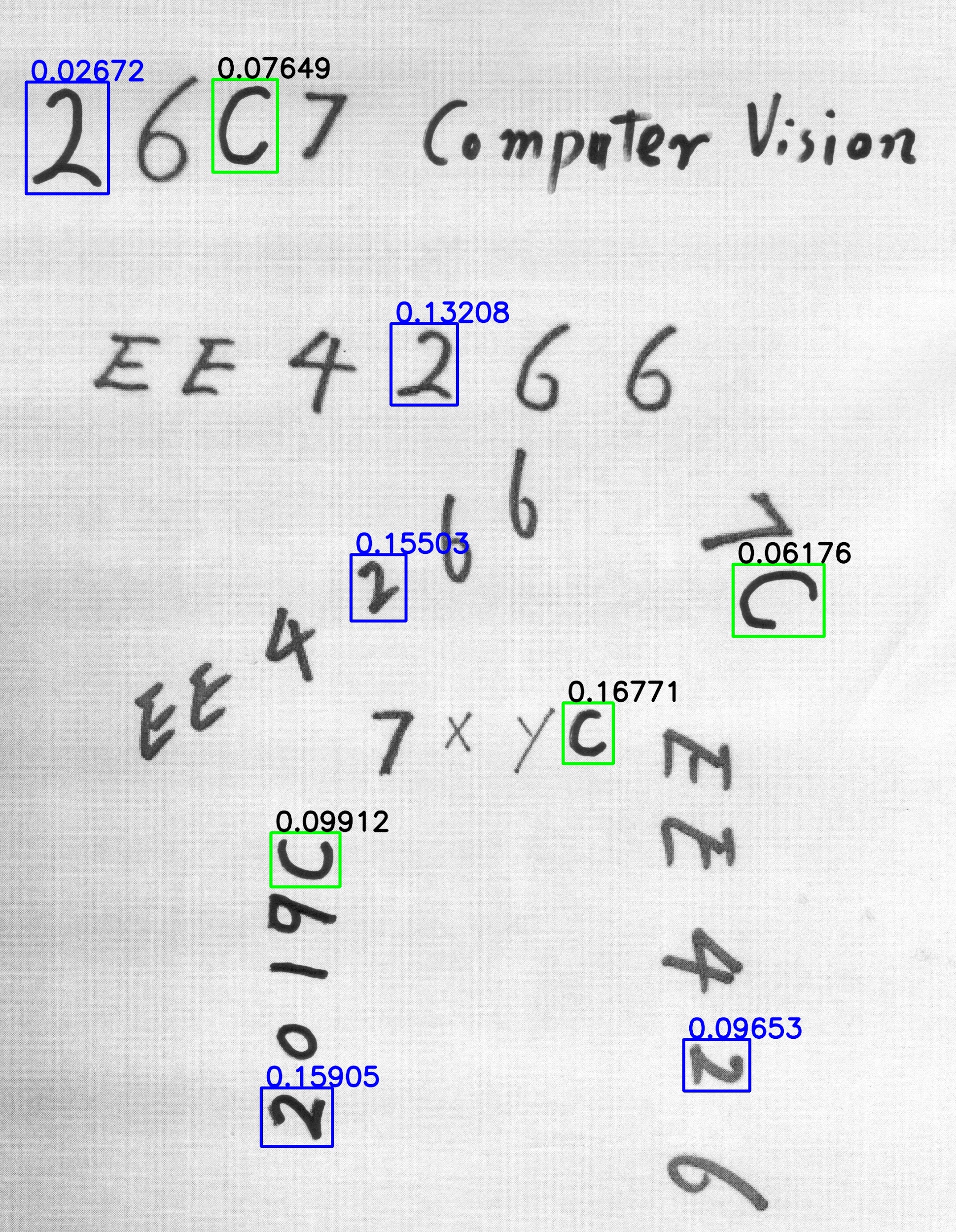
**Adjustments**

1. To improve the performance of the algorithm, additional preprocessing was performed.
2. Gaussian blur was applied to the templates (after Step 3 in Obtaining Contours) to reduce graining which caused the algorithm to perceive single contours as multiple discrete contours.
3. For ‘C’, cv2.erode() was used in addition to Gaussian blur to diminish the features of image and was found to be effective in removing the number of false positives (arising from ‘6’).
4. The thresholds for FFT were adjusted accordingly to accommodate the new adjustments.
   1. ‘2’ - 0.16
   2. ‘C’ - 0.17
5. The contours of ‘2’, ‘C’ and the sample before and after applying the adjustments were compared in the table below.

|  |  |
| --- | --- |
| Before | After |
|  |  |
|  |  |
|  |  |

1. After the modifications, the contours envelop a larger area of the character.
2. The inner loop of the character ‘6’ is absent and is believed to be the primary reason for the amount of false positives as it may have been perceived as an inverted ‘2’.

**Final Results**



1. A comparison between the overall results of the first and second attempt is recorded in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attempt | Accuracy (%) | Precision (%) | Recall (%) |
| 1 | 88.9 | 64.3 | 90.0 |
| 2 | 97.8 | 100.0 | 90.0 |

1. The modifications have successfully reduced the number of false positives compared to the first attempt. This is evident from the significant increase in precision.