1 Pre-Processing (phase)

We used OpenCV to read the image (prescription) and let the maximum width be 1000 because of speed.

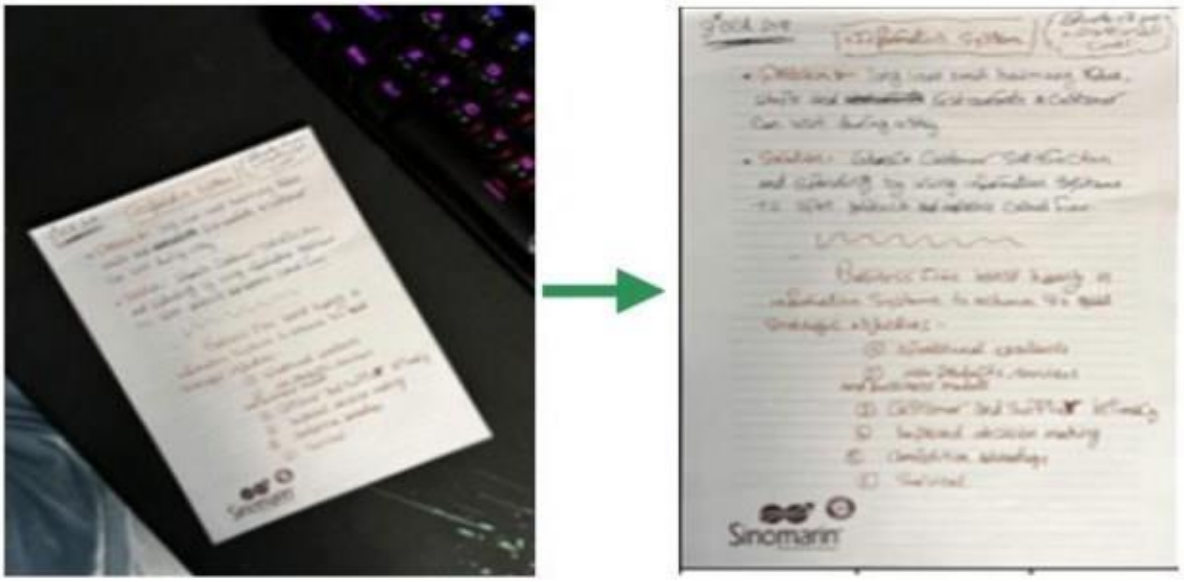
After that, we separated the background from the foreground (text) by threshold function, which is separated by a fixed number from 0 to 255 of a gray color degree.

2 Reading Prescription Problems

2.1 Warp Perspective (objects appear outside the frame of the paper)

A challenge arose, which is that in some pictures, some objects appear outside the frame of the paper, and therefore threshold also happens to them, and since they differ in color from the paper, this affects the separation process,
Therefore, they used the canny method to define the contours of the paper by drawing the borders of the shapes in the image, and then defining the corners of the paper by the bigger size of contour and cropping the image on the same area of paper

• Warp Perspective •



(Objects appear outside the frame of the paper)

2.2 Image enhancement (Can’t use fixed threshold only)

After that, a new challenge appeared while using a fixed threshold, which is the change of gray color degree from one picture to another because of the lighting of the photograph and the size and color of the writing line

To solve this problem, we used an adaptive threshold for several reasons:

1. Adaptability to Varying Illumination: One of the primary reasons for the success of adaptive Threshold over fixed thresholding is its ability to handle images with varying illumination conditions. Fixed thresholding applies a single threshold value to the entire image, which can lead to inaccurate results when the lighting conditions change across different parts of the image. Adaptive thresholding, on the other hand, adjusts the threshold value locally based on the pixel intensities in the vicinity, allowing it to handle variations in illumination more effectively.

2. Handling Non-Uniform Backgrounds: Images with non-uniform backgrounds pose a challenge for fixed thresholding methods. These methods often struggle to find an appropriate threshold value that can accurately separate the foreground from the background. Adaptive thresholding, by adapting the threshold value locally, can better handle non-uniform backgrounds and achieve more accurate results.

3. Robustness to Noise: Adaptive thresholding algorithms are generally more robust to noise compared to fixed thresholding methods. When applying a fixed threshold to a noisy image, the noise can interfere with the thresholding process and lead to false detections or inaccurate segmentation. Adaptive thresholding techniques incorporate noise reduction mechanisms

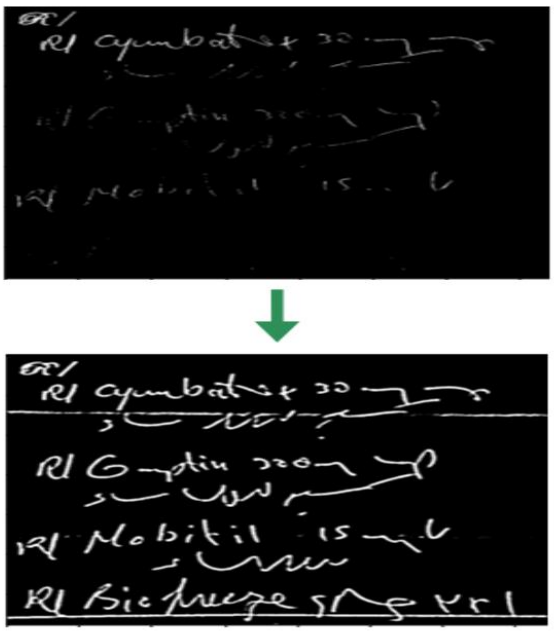
or statistical analysis within the local windowing approach, allowing them to mitigate the impact of noise and produce better results.

4. Localized Thresholding: Adaptive Threshold algorithms divide the image into small local regions or windows and compute the threshold value independently for each window. This localized approach enables the algorithm to adapt to the local characteristics of the image, such as texture, intensity variations, or object boundaries. By considering the local context, adaptive thresholding can achieve better segmentation results, preserve finer details, and handle complex image structures more effectively.

5. Increased Flexibility: Adaptive thresholding provides more flexibility in selecting the thresholding method based on the image characteristics or application requirements. There are various adaptive thresholding techniques available, such as mean, median, Gaussian, or Sauvola thresholding, each designed to handle specific image conditions. This flexibility allows users to choose the most appropriate thresholding method for their specific application, resulting in improved performance and accuracy.

- **Image enhacement -**

- Cannot use fixed thresholds only.
- Thus, we solve the problem of external factors affecting the image.



3 Segmentation of the interested region

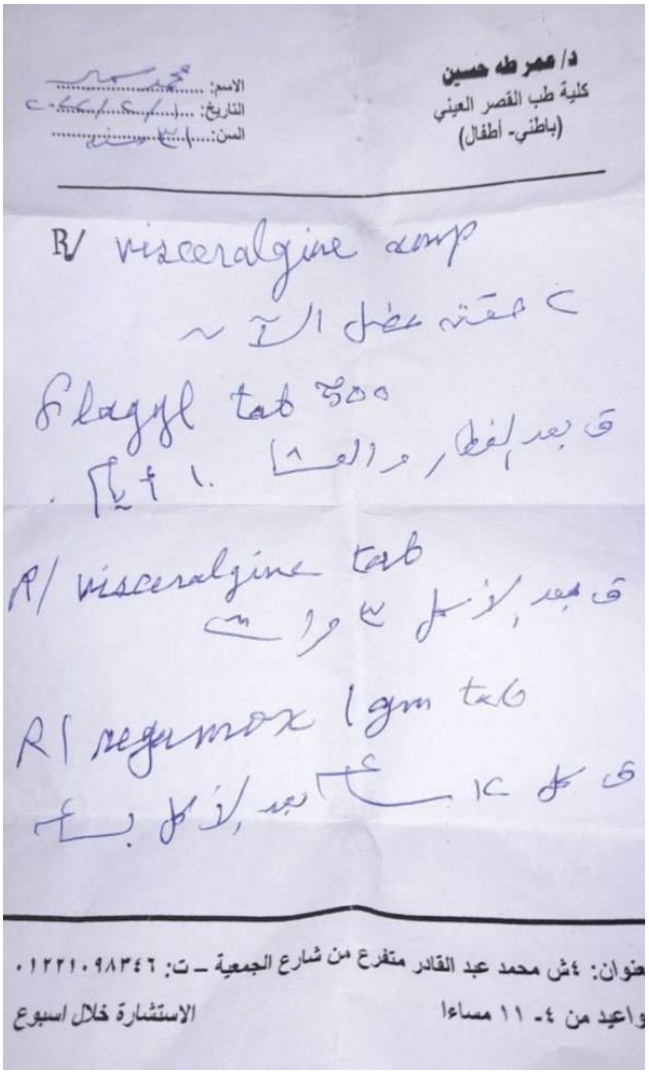
3.1 Remove Header and Footer of Prescription

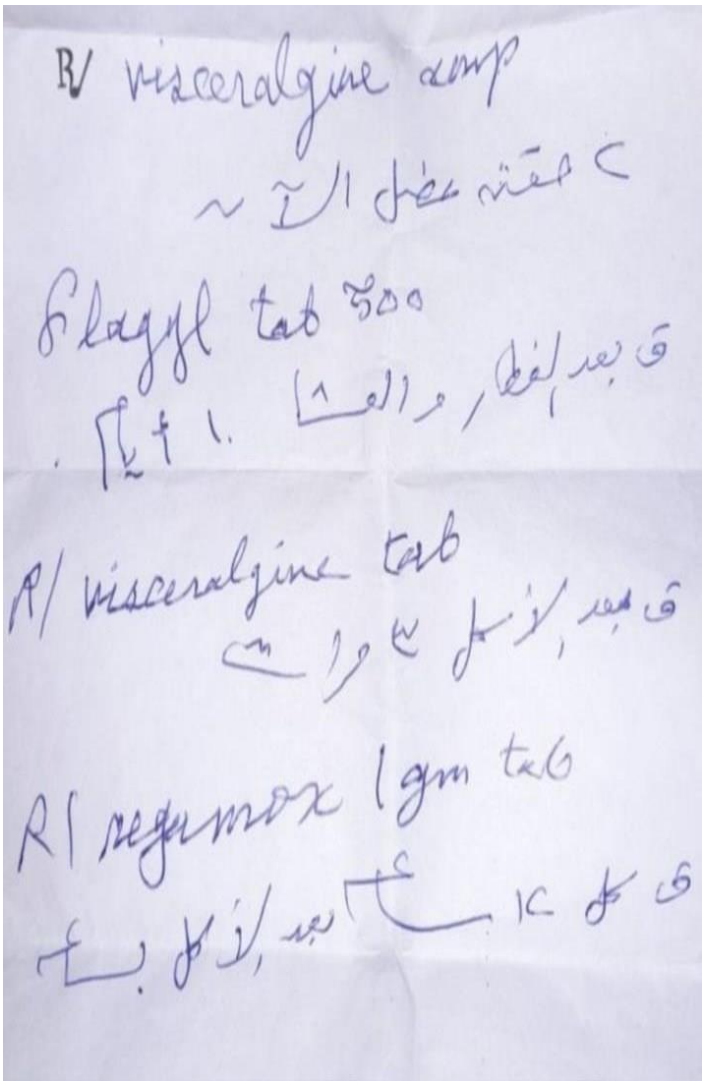
We had to remove the header and footer from the paper in order to reduce the model's distraction from reading unimportant words.

To remove the header and footer we made dilation with very small Kernel of ones to accentuate features and increase the object area, and we added Gaussian Blur and Threshold to improve the accuracy of text to help with the dilation process, and we used the FindContours function to define the contours of the texts, then put each Contour whose width is greater than 80% of paper's width to the list because we sure that is no word his width is greater than 80% of paper's width, so it is a horizontal line.

Then we cut paper from up to the first line in the list and from down to the last line in the list because of could be lines in the paper another header and a footer.

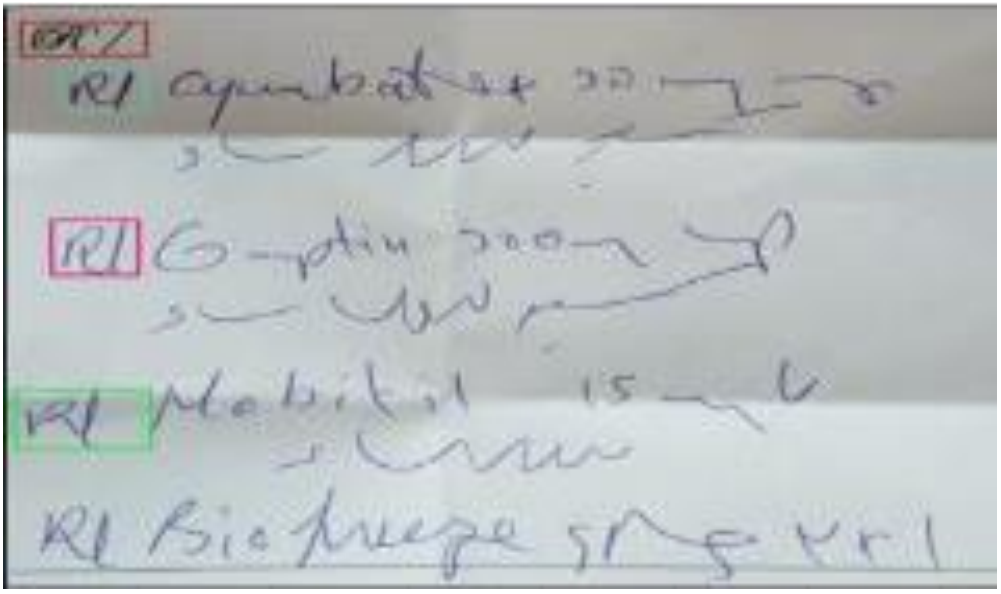
Remove Header & Footer





3.2 Remove “R/” symbol

During the test, we found that in some images, the "R\" symbol was written indicating something, but it was not included in the name of the medicine, so this became the new challenge.



After much thought, we used the idea of Character detection through the CNN to identify the letter "R", so we used a pre-train model on the character recognition, but the idea failed because Often the "R" character is combined with "\" so that it cannot be recognized correctly

Instead of taking much time and resources to make a specific dataset, we thought that there is no medicine name consisting of two characters or one character, so we made dilation with a very small kernel to recognize small areas, but it will search only in the first third of the image because the "R\" will be at the beginning of the line only

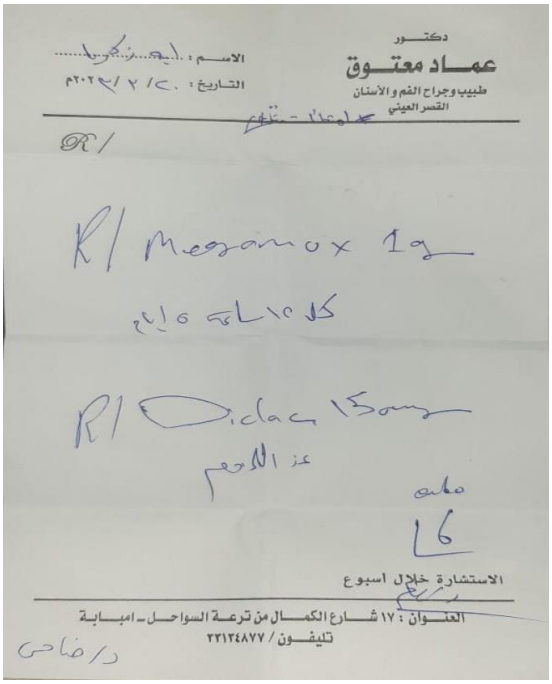
We also found that the image that contains "R\" at the beginning of each line has the "R\" printed at the beginning of the paper, so we took the space of the

printed symbol, and then we will take only the "R\" in the first third of the paper, and its area does not exceed twice the area of the printed symbol and the distance between it and the text that followed by within the same limits as the mean distance between words

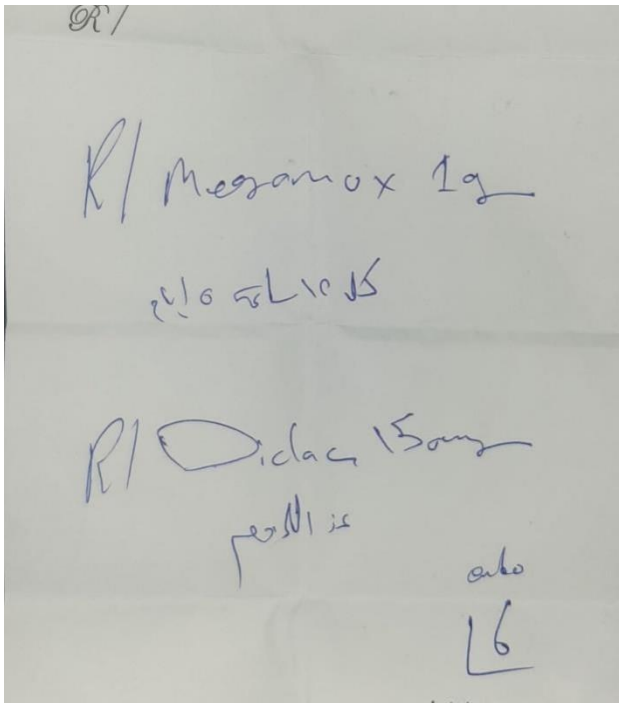
In this way, we have also solved part of the dispersion on the model as well, and we have reduced the problem of merging the words that we will display in the next slide.

In addition, here we have come to a significant part of the pre-processing, we have removed some of the things that could distract the model, and now we have only the important part of the medical prescription, and we are ready to start the part of identifying the names of the drugs, starting from the slide after the next

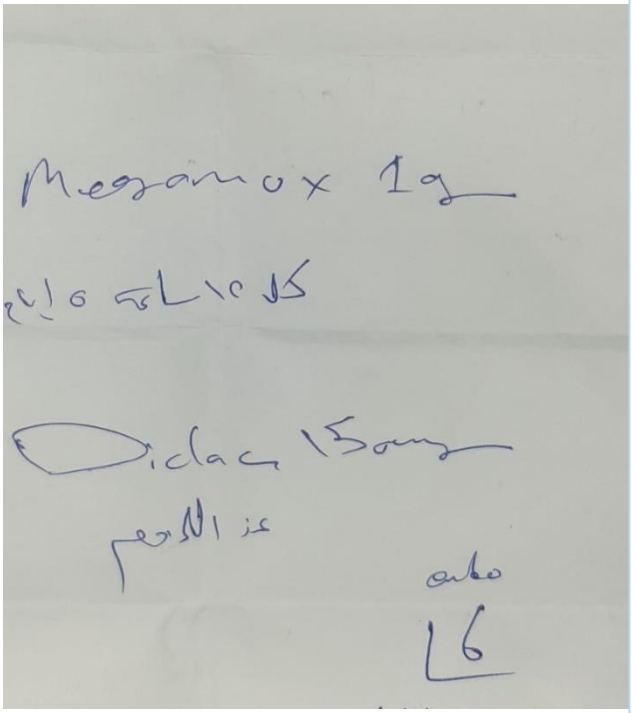
Segmentation of the interested region



a) The original prescription



b) Remove Header and Footer



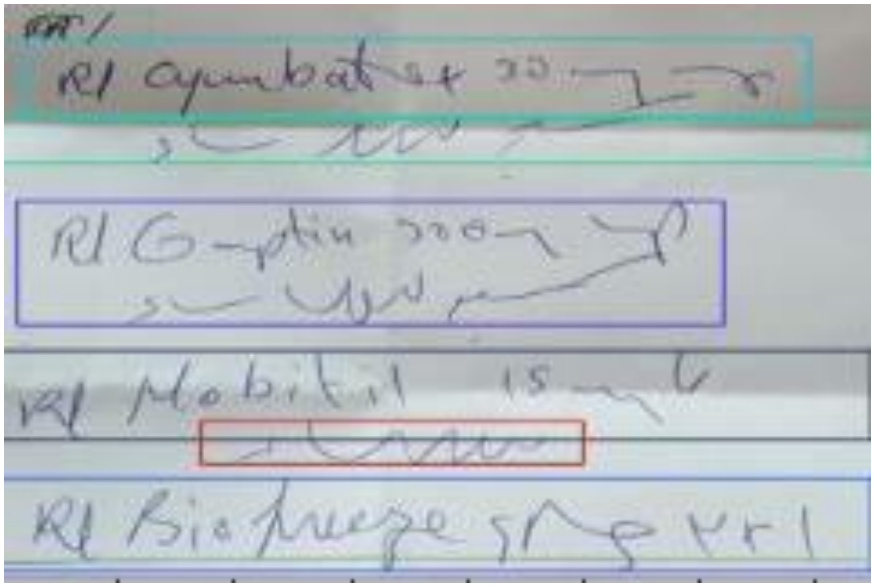
c) Remove "R/" Symbol

4 Segmentation of text

Then we make dilation with Kernel of ones to accentuate features and increase the object area, and we added Gaussian Blur to improve the accuracy of text to help with the dilation process, and we used the FindContours function to define the contours of the texts

Thus, we have entered the line segmentation stage, so we used a large kernel to define each line in the image and define each line with a frame by the contours

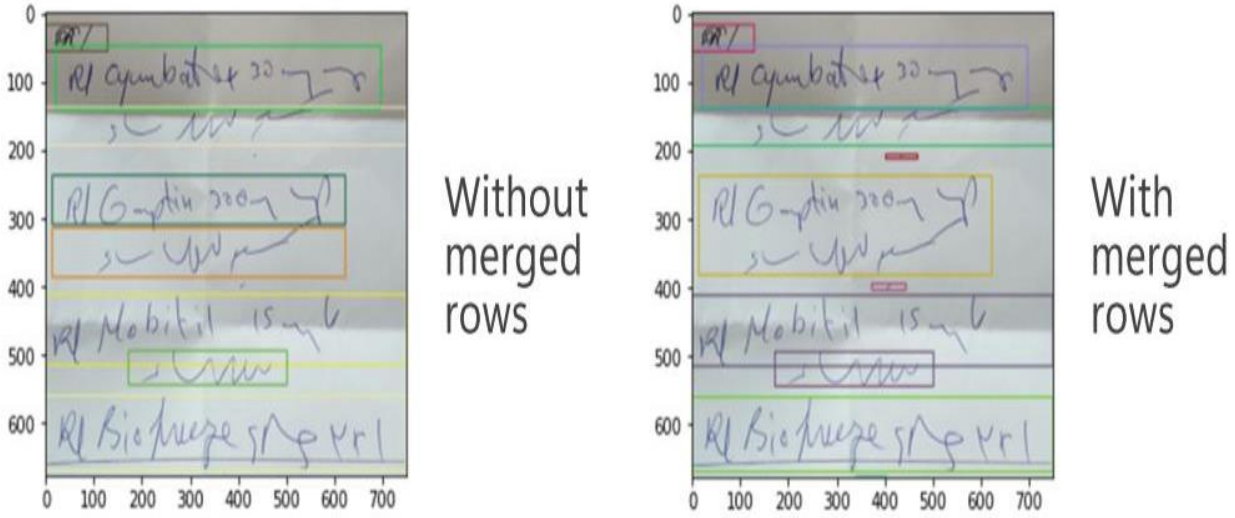
We faced a new challenge, which in some pictures puts borders on two lines as if they were one line, and this is because of the difference in the distance between each line and length of each line



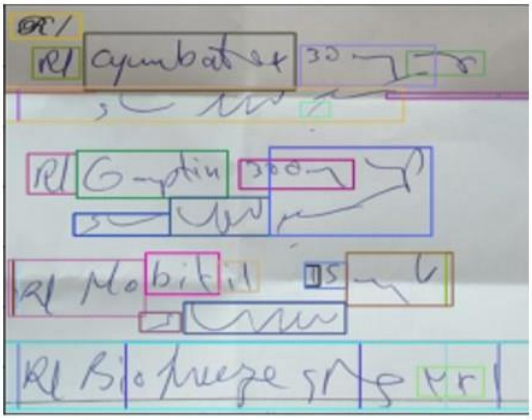
And we tried to solve it with a set of steps, the first of which is the work of function calculating the average line length and function calculating the average spacing between the lines, and we stipulated that each line should not exceed twice the mean length of the lines, and if this happened, then we would divide the line

length into two halves, but we found that in some other images, the name of a medicine is written larger than the rest of the medicine in the paper.

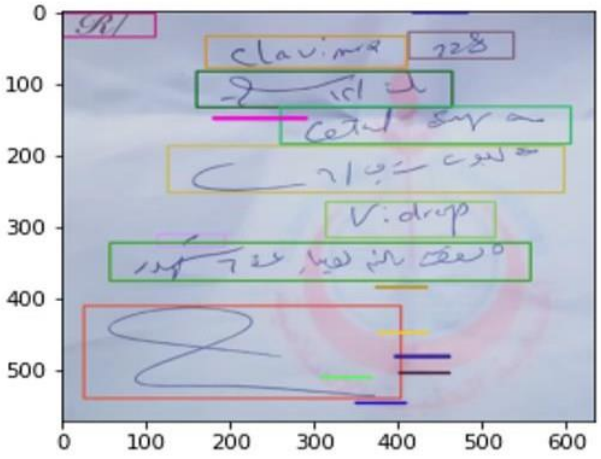
In this case, the line length is greater than the Double mean length of the lines, the line is divided into two lines, so we overlooked the idea of dividing the line into two lines so as not to lose some medicine names, as it will also identify the words inside each line through the word segmentation



After that, we made dilation by a small kernel inside each line in the image to define each word in the lines to avoid merging of words, and we put borders around each word and we removed noise below the mean word length and cut the paper to images of word

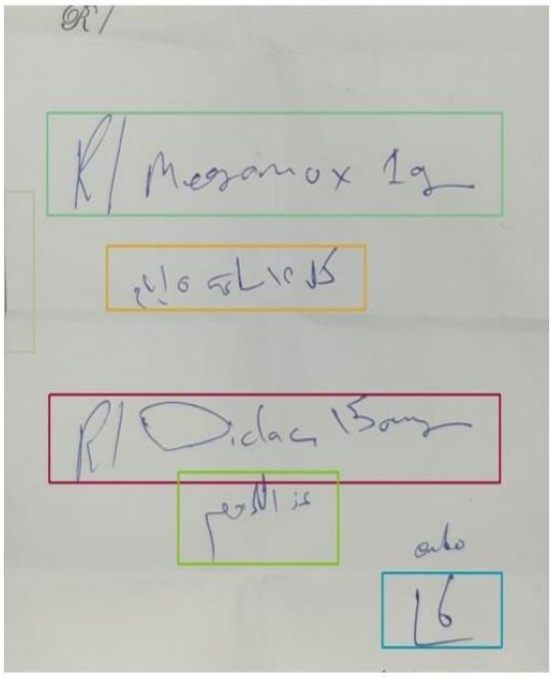


Overlap

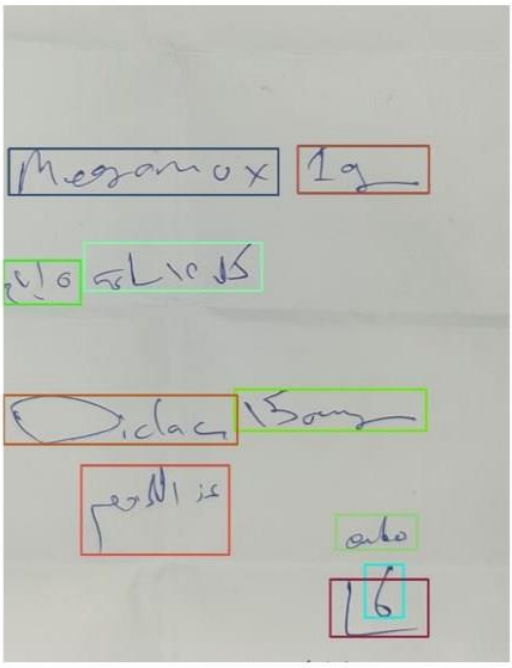


Noise

Moreover, we are almost done with pre-processing's part, we removed the noise, we identified all the words in the picture, we put borders on them, and we are ready to recognize the name of the drug that the patient chooses from the picture



Perfect Line Segmentation



Perfect Word Segmentation