Assignment 3: Handwriting characters extraction project documentation

Lines extraction

Lines extraction is implemented by the extract_text_lines function which takes as input the resized image (which is 18% of the original size of the image) and saves the extracted lines into the given output directory

The lines extraction is implementing using the following steps:

1. Convert image to grayscale for faster processing
2. Blur the image using 5x5 median blur kernel in order to reduce any possible noise (especially salt-and-pepper noise)
3. Apply adaptive threshold in order to highlight the handwritten text from the image with the following parameters

   maxValue = **255**, meaning if pixel value is more than the threshold value then it will become a white pixel otherwise all the other pixels will become black

   adaptiveMethod = **ADAPTIVE_THRESH_GAUSSIAN_C** – threshold value is the weighted sum of neighborhood values where weights are a Gaussian window which will have the effect of not being sensitive to detect finer edges

   thresholdType = **THRESH_BINARY_INV**, meaning that the given that it will invert the colors which will have the effect of converting all the handwritten text in white and the background in black

   blockSize = **5**, meaning Gaussian window size which determines pixel neighborhood used to calculate the threshold value

   C = 5, a bias constant that is subtracted from the weighted mean to get the final pixel value
4. Create a rectangular kernel for dilation with the shape of (16,2), meaning that the dilation will be made horizontally
5. Dilate the thresholded image with the created kernel which will had the effect of expanding the handwritten characters area horizontally, in order for any void pixels within characters and any empty space between characters will be filled up. This will 'unite' the characters' areas from within the same line in order to form a plain horizontal and considerably irregular rectangle line which covers the entire handwritten text line
6. Find the contours of the created irregular 'lines' of white pixels
7. Iterate through the found contours in order to get the bounding box coordinates of all the found lines. These coordinates will help extracting the handwritten text lines regions from the original image

Characters extraction

Characters extraction is implemented by the extract_text_chars function which takes as input **the original image** and saves the extracted lines into the given output directory

The characters extraction is implemented using the following steps:

1. Convert image to grayscale for faster processing
2. Blur the image using 7x7 median blur kernel in order to reduce any possible noise (especially salt-and-pepper noise)
3. Apply adaptive threshold in order to highlight the handwritten text from the image with the following parameters

    maxValue = 255, meaning if pixel value is more than the threshold value then it will become a white pixel otherwise all the other pixels will become black

    adaptiveMethod = ADAPTIVE_THRESH_MEAN_C – threshold value is the mean of neighborhood area which will have the effect of being very sensitive to detect finer edges. This will result in characters will will be more filled

    thresholdType = THRESH_BINARY_INV, meaning that the given that it will invert the colors which will have the effect of converting all the handwritten text in white and the background in black

    blockSize = 7, meaning pixel neighborhood used to calculate the threshold value. The window is bigger because the input image is bigger

    C = 11, a bias constant that is subtracted from the weighted mean to get the final pixel value
4. Create a small circular kernel for the dilation operation with the shape of (3,7), meaning that the dilation will stretch more vertically than horizontally
5. Dilate the thresholded image with the created kernel which will have the effect of emphasizing more the handwritten characters and their edges
6. Find the contours of all the distinct (now emphasized characters) from the image
7. Iterate through the found contours in order to get the bounding box coordinates of all the found characters. These coordinates will help finding the handwritten characters regions from the original image in order to extract each individual handwritten character from the image