## Task 2: What did you learn from the code above? Analyze your understanding in your own words.

Initially, in the code, all the necessary libraries needed for image processing are imported. Ex- cv2 and matplotlib.

# Image Processing

Firstly, an image of a braille script is loaded, and it is converted to a grayscale image, then the converted image is processed using the thresholding concept.

1. Here an image called “alpha.jpg” is loaded.
2. On using the print statement, it prints a matrix instead of the image because we have used the cv2 imread() function which reads the image and generates the below output.

Then, we use the print function to print the shape of the image which is (640,640,3) (Height, weight, number of channels). The number of channels for a grayscale image =3 as it is a color image.

1. Finally, we used the imshow() function to display the image that was loaded.

Graphical user interface, application, Word

Description automatically generated

Graphical user interface

Description automatically generated

1. Next, we use cv2.cvtColor() function to convert the actual image to a grayscale image.

Note: Grayscale image is usually an 8-bit image with each pixel having 256 combinations of shades of gray. A color image is usually a 24-bit image with 8-bits of Red, 8-bits of Green, and 8-bits of blue data.

1. Then, we use the print function to print the shape of the image which is (640,640)

(Height, weight, number of channels). The number of channels for a grayscale image =0.

1. Below is the black and white image of the actual image.

Graphical user interface, application

Description automatically generated

1. For image processing, to identify the text in the image, we try to enhance the image using thresholding.

Note: The thresholding methods replace each pixel in an image with a black pixel if the image intensity {\displaystyle I\_{i,j}}is less than a fixed value called the threshold{\displaystyle T}, or a white pixel if the pixel intensity is greater than that threshold.

1. Adaptive threshold for the blurred image is calculated, which calculates the threshold for smaller regions.
2. Then, median blur replaces the middle element of the image with the median of all the pixels in the kernel and the threshold is calculated.
3. Then, the gaussian blur smoothens the sharp edges in images and the threshold is applied.

Graphical user interface, application

Description automatically generated

1. We use the feature extraction technique (ORB- Oriented FAST and Rotated Brief - **ORB** is an efficient alternative to SIFT or SURF algorithms used for feature extraction, in computation cost and matching performance) on the image to convert the image to a set of coordinates of each dot in the image. Here we only get the first five coordinates of the points.
2. Then, we plot the points.

Graphical user interface, text, application, Word

Description automatically generated

1. We now compute the minimum and maximum coordinates.

Graphical user interface, application

Description automatically generated

1. For the above image perspective transformation is applied to align the text in the image.

Graphical user interface, application

Description automatically generated

1. Once the image is perfectly aligned, we slice it into pieces that make sense.

Graphical user interface, text, application

Description automatically generated

# Video Processing

1. For Video processing, we are initially loading a motion video “body.mp4” and printing the number of frames in the video. Here the total frames = 581.

Graphical user interface, website

Description automatically generated

1. Now, we convert the color video to a grayscale video.
2. Then, we calculate the difference between each frame to analyze the difference in the activity in each frame.
3. Then each frame with different amps is plotted as shown below.

Graphical user interface, application

Description automatically generated

1. Below is the “display\_images()” function used to plot the frames.

Graphical user interface, text, application, email

Description automatically generated

1. Now, we have to set a threshold value and frames with several changes above the threshold value.

Graphical user interface, application

Description automatically generated

1. As the minimum length is 30, only the consecutive frames longer than 30 frames are extracted. And display the image.

Graphical user interface, application, Word

Description automatically generated

1. Finally, the color of the image is converted as OpenCV uses BGR colors while matplotlib uses RGB colors.

Graphical user interface, application, Word

Description automatically generated

1. **Optical flow is a pattern of motion of an image between two consecutive frames caused by the movement of an object or camera. The dense optical flow** of the frames is calculated as shown below.

Graphical user interface, text, application, email

Description automatically generated

1. Finally, optical flow is converted to Hue-saturation value. And the frames are displayed. The color in the frames indicates right and left motion. The green color motion is towards the left and the blue color indicates motion towards the right.

Graphical user interface, text, application, chat or text message

Description automatically generated