**Line follower:**

The main idea of a line follower is to take an image as input and then extract features related to lines in the image to determine direction, relative position, and size of the line to be followed. The algorithm used consists of several steps that optimize computations of the detection method and achieves high accuracy.

First of all, the main library used to store and manipulate images in python is open cv. Since the image will be a part of a video we expect it to have some noise related to movement. The effect of this noise could be alleviated using a bilateral filter and Gaussian blur which we applied to input image. Several testing trials proved that the blur kernel of size(15x15) achieved the best results. The next step of preprocessing the image is to convert it from the RGB/BGR color space to the HSV color space. The reason we use HSV color space for color detection/thresholding over RGB/BGR is that HSV is more robust towards external lighting changes. This means that in cases of minor changes in external lighting (such as pale shadows, etc.) Hue values vary relatively lesser than RGB values. So, in short, HSV images makes it easier to select a color region because it separates color info from intensity info.

The next step is to determine a color range in the HSV color space and look for any pixel value in that range in the image, therefore extracting only the part of the image with the line. The optimum range(s) were determined empirically through trail. The output of the last step is a binary mask that is only true for pixel values that are the same color as the line.

The Last two steps are edge extraction and line detection. For the first part the image is passed to a Canny filter that returns the image with only the edges. The last and final part is to use the edges found to detect lines using a Hough transform. The idea behind a Hough Transform is that lines can be expressed in polar coordinates as instead of and then points on the same line can be determined.

//final output here (angle, thickness, distance from center)

Finally, the output of this algorithm is published as a ROS node to communicate its results.