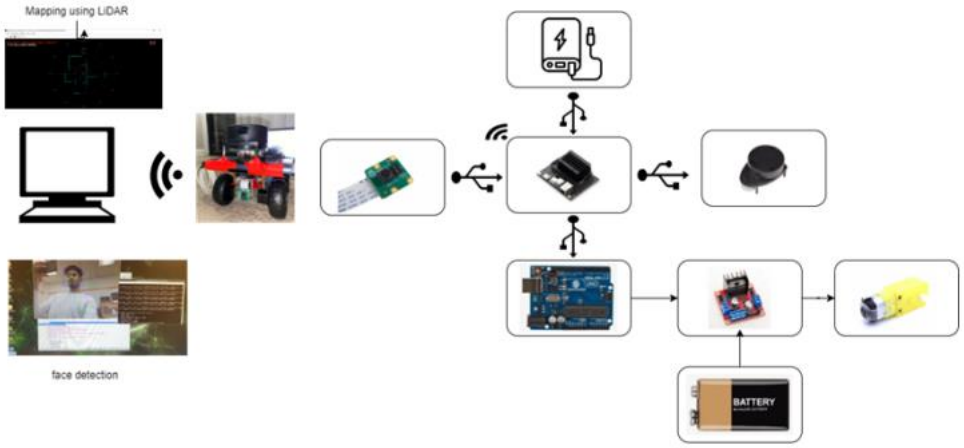
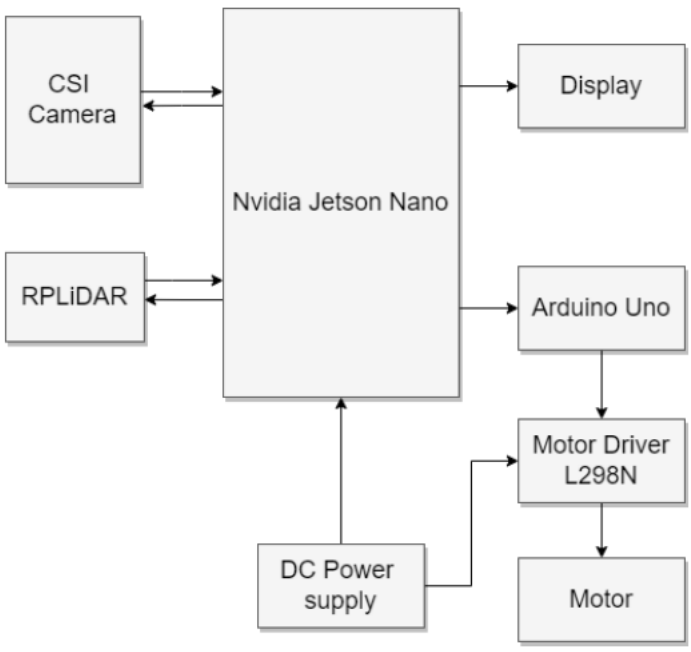
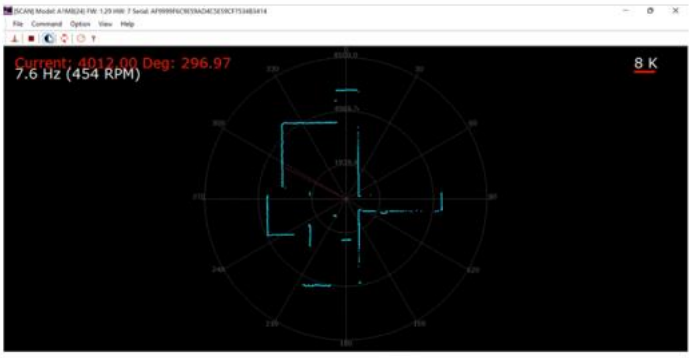
METHODOLOGY



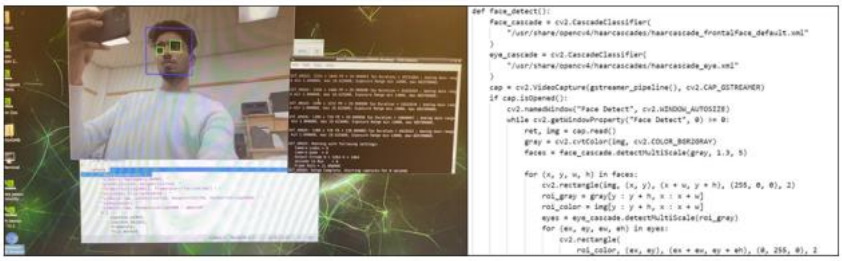
The proposed system aims to aid the rescue officials in the search and rescue operations. Portability is important aspect in such robots. The robot is powered using a 10000 mAh power bank. The NVIDIA Jetson Nano 2GB module is the backbone of the system. As shown in figure, it takes inputs from the RPLiDAR A1M8 as well as the Raspberry Pi V2 camera, processes the information and displays the output. The robot is driven by four motors which are in turn driven by the motor driver L289N. The Arduino UNO acts as an intermediate between Jetson Nano and the motor driver.



When the robot enters an environment, the LiDAR detects the unknown space and maps it. As shown in figure, the RPLiDAR maps the environment through 360 degrees with a range of 6 meters. The circle visible in the snapshot represents the 360 degree scanning. The green lines seen in figure represent the obstacles in the environment. The green lines change with the change in the environment. As the robot moves towards a different direction, it keeps changing path thus we can see a variation in the mapping.



The Raspberry Pi V2 camera is responsible for the face detection aspect in the system. The robot captures a video in real-time and sends it to the Jetson Nano for processing. The frames captured by the camera are first converted to grayscale. Each image frame from the camera from the live stream is processed and checks for face detection. The code uses pre-trained Haar Cascade implementations stored in the OpenCV library. The code uses the GStreamer pipeline to create an interface between the camera and the OS. The face detection can be seen in figure. It can be seen in the figure that the eyes and face of an human is deteted. Since, the color does not determine the facial features, it avoids computational overheads and enhances performance.



The Search and Rescue robot uses the RRT algorithm. The rapidly exploring random tree (RRT) algorithm is a technique that uses a random space filling tree to swiftly search non-convex, high-dimensional space. The tree samples will be picked at random from the search space, which will extend to include enormous areas that have yet to be explored. RRT can also be called as a Monte Carlo method for searching big graph voronoi areas. The map of the environment is divided into an obstacle region

and obstacle-free region. The start position of the robot is represented with red dot in the map. The goal region of bot in the environment is represented in green dot. The algorithm continues the iterations until the robot reaches the goal.