|  |  |
| --- | --- |
| DAY – 51 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID | |
| Remark of training Supervisor: | |
| Introduction to ML:   * Machine Learning (ML) is a subset of artificial intelligence (AI) that focuses on enabling machines to learn from data and improve their performance without being explicitly programmed. It involves the development of algorithms and models that can automatically analyze and interpret patterns in data, extract meaningful insights, and make accurate predictions or decisions. * The core idea behind machine learning is to train algorithms using labeled or unlabeled data to recognize patterns and relationships   Signature of industry Supervisor | |

|  |  |
| --- | --- |
| DAY – 52 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID | |
| Remark of training Supervisor: | |
| Description of ML:  There are several types of machine learning algorithms:   * Supervised Learning: In supervised learning, the algorithm is trained on a dataset where both the input features and their corresponding labels or outputs are provided. * Unsupervised Learning: Unsupervised learning deals with unlabeled data, where only the input features are available. The algorithm learns to find patterns, structures, or relationships in the data without any predefined labels. * Reinforcement Learning: Reinforcement learning involves training an agent to interact with an environment and learn optimal behaviors through trial and error.   Signature of industry Supervisor | |

|  |  |
| --- | --- |
| DAY - 53 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID: | |
| Remark of training Supervisor: | |
| Description of ML:   * Real-time Decision-Making: ML algorithms can make decisions and predictions in real-time, enabling organizations to respond quickly to changing conditions and make informed decisions in dynamic environments * Continuous Learning and Improvement: ML models have the ability to continuously learn and improve their performance as new data becomes available. They can adapt to changing patterns and update their knowledge to make more accurate predictions * Data-Driven Decision Making: ML enables organizations to make data-driven decisions by leveraging insights and predictions derived from large datasets.     Signature of industry Supervisor | |

|  |  |
| --- | --- |
| DAY –54 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID: | |
| Remark of training Supervisor: | |
| Introduction of OpenCV:   * OpenCV open source computer vision library is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision application and to accelerate the use of machine perception in the commercial product. Being an Apache 2 licensed, OpenCV makes it easy for business to utilize and modify the code. * The library has more than 2500 optimize algorithm, which includes a comprehensive set of both classic and state of the art computer vision and machine learning algorithms.     Signature of industry Supervisor | |

|  |  |
| --- | --- |
| DAY - 55 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID: | |
| Remark of training Supervisor: | |
| Description of OpenCV:  Some key features and functionalities of OpenCV include:   * mage Processing: OpenCV provides a wide range of image processing functions, such as resizing, cropping, filtering, thresholding, edge detection, and morphological operations. * Video Analysis: OpenCV supports video input and processing, enabling tasks such as video capture, frame extraction, video compression, and object tracking in videos. * Object Detection and Recognition: OpenCV offers pre-trained models and algorithms for object detection, recognition, and tracking. It includes popular object detection methods like Haar cascades and HOG (Histogram of Oriented Gradients).     Signature of industry Supervisor | |

|  |  |
| --- | --- |
| DAY - 56 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID: | |
| Remark of training Supervisor: | |
| Functionality of OpenCV:   * Image and Video I/O: OpenCV allows reading and writing images and videos in different formats. It provides functions for image and video capture from cameras, file systems, or other sources, as well as the ability to save processed images or videos. * Image Processing: OpenCV includes a comprehensive set of image processing functions. It offers operations such as resizing, cropping, rotation, flipping, filtering opeartions * Feature Detection and Extraction: OpenCV provides algorithms for detecting and extracting features from images.   Signature of industry Supervisor | |

|  |  |
| --- | --- |
| DAY - 57 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID: | |
| Remark of training Supervisor: | |
| Introduction of Haarcascade:   * Haar cascade is an algorithm that can detect objects in images, irrespective of their scale in image and location. * This algorithm is not so complex and can run in real-time. We can train a haar cascade detector to detect various objects like cars, bikes, buildings, fruits, etc. Haar cascade uses the cascading window, and it tries to compute features in every window and classify whether it could be an object.     Signature of industry Supervisor | |

|  |  |
| --- | --- |
| DAY - 58 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID: | |
| Remark of training Supervisor: | |
| In compay we are implemented “Traffic Light Detection Appliction” using OpenCV below steps are used to implement this project:  1. The program starts by importing the necessary libraries: `cv2` for computer vision operations and `numpy` for numerical operations.  2. The `detect\_traffic\_lights` function is defined, which takes an input image as a parameter.  3. Inside the function, the input image is converted from the BGR color space to the HSV color space using `cv2.cvtColor` function. HSV (Hue, Saturation, Value) color space is often preferred for color-based image processing tasks.  4. Color ranges for red, yellow, and green are defined in the HSV color space using lower and upper threshold values. These ranges determine the valid color ranges for each traffic light color. 5. The HSV image is thresholded using `cv2.inRange` to create binary masks for each color. The `inRange` function converts the pixels within the specified range to white (255) and the rest to black (0), resulting in binary images where the colors of interest are white and the rest are black.    Signature of industry Supervisor | |
| DAY - 59 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID: | |
| Remark of training Supervisor: | |
| Steps are continued:  6. Contours are then found in the binary masks using the `cv2.findContours` function. Contours represent the boundaries of connected regions in an image.  7. To filter out small contours that are likely to be noise or undesired detections, a minimum contour area threshold is defined. Contours with an area smaller than this threshold are excluded. 8. The filtered contours are then processed individually. For each contour, a bounding rectangle is calculated using `cv2.boundingRect`, which returns the coordinates (x, y) of the top-left corner of the rectangle, as well as its width (w) and height (h).  9. The bounding rectangle is drawn on the original image using `cv2.rectangle` with the appropriate color and thickness. The text label corresponding to the detected color (e.g., "Red", "Yellow", "Green") is added using `cv2.putText`.  10. Finally, the processed image with bounding rectangles and labels is returned.    Signature of industry Supervisor | |

|  |  |
| --- | --- |
| DAY - 60 | DATE: |
| TIME OF ARRIVAL: | TIME OF DEPARTURE: |
| Dept/Division: | Nature of work: |
| Name of Supervisor, designation & email ID: | |
| Remark of training Supervisor: | |
| Steps are continued:  11. The program then loads an input image using `cv2.imread`.  12. The `detect\_traffic\_lights` function is called with the input image, and the result is stored in the `result` variable.  13. The resulting image with bounding rectangles and labels is displayed using `cv2.imshow`.  14. The program waits for a key press (`cv2.waitKey(0)`) and then closes all open windows (`cv2.destroyAllWindows()`). This program processes the input image by converting it to the HSV color space, thresholding it based on color ranges, finding contours, filtering out small contours, and drawing bounding rectangles and labels for the detected traffic lights.    Signature of industry Supervisor | |