Case Study

# “Tesla’s autonomous EVs”

**-Chip used:** NVIDIA Drive AGX Pegasus AI computer

**-Sensors used:** Cameras (front, sides, and rear), Ultrasonic sensors (bumpers), Radar (front)

**-Common Algorithms:**

1. **Convolutional Neural Networks (CNNs):** These are algorithms used for image recognition and classification and designed to process and analyze large amounts of visual data in real-time. For example, *LeNet-5 is used for traffic sign detection.*
2. **Object Detection:** Object detection algorithms identify and track objects in the vehicle’s field of view, such as other vehicles, pedestrians, and road signs. The techniques used are:
   1. Histogram of Oriented Gradients (HOG):
      1. Operates on *grayscale images*.
      2. Involves computing the image gradients in both the horizontal and vertical directions.
      3. The image is divided into small cells.
      4. The gradients are used for each cell to calculate the histogram of gradient orientations.
      5. The cell histograms are grouped into larger blocks, and Normalization is applied.
      6. This captures the distribution of gradient orientations across the image.
   2. You Only Look Once (YOLO):
      1. *Object detection in a single forward pass* of the neural network makes it significantly faster.
      2. Divides the input image into a grid and predicts bounding boxes and class probabilities.
      3. Each grid cell is responsible for predicting bounding boxes and class probabilities for objects in the cell.
      4. Each grid cell predicts multiple bounding boxes along with confidence scores.
      5. For each bounding box, YOLO predicts class probabilities.
      6. YOLO uses non-maximum suppression to refine the final set of predictions, which removes redundant bounding boxes based on their confidence scores.
3. **Optical Flow:** Optical flow algorithms are used to estimate the movement of objects in the vehicle’s field of view, which helps the system decide how to control the car. The algorithms used are:
   1. Lucas-Kanade:
      1. Assumes that the motion between consecutive frames is constant within a local neighborhood.
      2. Computes the spatial gradients of the image.
      3. Calculates the temporal gradient, representing the intensity change between consecutive frames.
      4. Sets up a system of linear equations based on the spatial and temporal derivatives, which is then solved to estimate the optical flow.
      5. Often, a Gaussian window is applied to give more weight to the central pixels of the neighborhood, assuming that they are more reliable for motion estimation.
      6. The output of the Lucas-Kanade method is an optical flow vector for each pixel in the image, representing the estimated motion in the x and y directions.
      7. Assumes that brightness remains constant within the local neighborhood, “Brightness Constancy.”
   2. Horn Schunck:
      1. Global optical flow estimation method that considers the entire image rather than local regions.
      2. Assumes that the optical flow field is smooth across the entire image. This is known as the "smoothness constraint."
      3. Formulates the optical flow problem as a variational optimization problem. The goal is to minimize an energy function, including data matching (brightness constancy) and the smoothness constraint.
      4. The Euler-Lagrange equation is derived from the variational formulation, and its solution provides the optical flow field.
      5. Typically, it involves pre-processing the images with Gaussian smoothing to reduce noise and enhance the accuracy of the flow estimation.
4. **Lane Detection:** Lane detection algorithms identify and track the road lanes in the vehicle’s field of view. Some algorithms are:
   1. Hough Transform:
      1. Works by transforming the Cartesian coordinate space (x, y) into a parameter space, often represented as (ρ, θ) for line detection.
      2. In the (ρ, θ) space, each line in the Cartesian space corresponds to a sinusoidal curve. The parameters ρ and θ represent the distance from the origin to the closest point on the line and the angle of the regular line to the origin, respectively.
      3. An accumulator array keeps track of the intersections or peaks in the (ρ, θ) space.
      4. For each edge point in the original image, the Hough Transform *votes for possible lines it could belong to in the parameter space.*
      5. The intersection points in the accumulator array with the most votes correspond to the parameters of the detected lines.
      6. A threshold is often applied to the accumulator array to filter out noise and irrelevant intersections, keeping only the significant peaks.
5. **Semantic Segmentation:** Semantic segmentation algorithms segment the image into different regions, each representing a different object or road feature. This information can then be used to make driving decisions, such as avoiding obstacles or following a specific lane.
6. **Localization:** The EV, at all times, must have a reasonably accurate idea about its position in a given map or environment. It can only decide what to do next if it knows where it is. To localize itself, it has access to relative and absolute measurements containing information about its position. It gets this information from its sensors. The sensors give it feedback about its movement and the environment around the EV. Given this information, the Car has to determine its location as accurately as possible. But we get uncertainty in both the robot's movement and sensing. The uncertain information needs to be combined optimally by using algorithms such as the Extended Kalman filter algorithm:
   1. Extended Kalman filter algorithm: The Extended Kalman Filter (EKF) is an extension of the traditional Kalman Filter and is commonly used for state estimation in nonlinear systems. The standard Kalman Filter assumes linear dynamics and Gaussian noise, but the system dynamics are nonlinear in many real-world applications. The EKF addresses this limitation by linearizing the system at each time step.