|  |  |
| --- | --- |
|  | **Department of Computer Science**  **University of Management and Technology** |

**Lane Detection System Using MATLAB**

|  |  |
| --- | --- |
| Course Name: | Digital Image Processing |
| Section: | V3 |
| Teacher: | Dr Asma Naseer |
| Assignment No.: | Milestone 3 |
| Deadline: | 26 april 2020 |
| Student Name | Muhammad Hamza Aslam |
| Student ID | F2017266041 |

Note: all pre-processing are describe bellow on some research base or by reading some papers

Algorithm: I design algorithm but not implemented yet

**Data Set:**

Video taken from

<https://github.com/sujaybabruwad/LaneDetection>

**Pre Processing:**

* First we import video file which is in .mp4 format
* Then we load file origin and its interest variables
* Then we create our output video also set video frame rate then show on screen

**Algorithm:**

* Then I use Gaussian filter for sharpen an image
* Then I have to show lane detection so I use white and yellow color for lane detection
* I use threshold for saturation for both white and yellow
* Create mask for histogram for both white and yellow
* Then I use canny filter for edge detection for both yellow and white mask
* I use roipoly function to extract region of interest (filtering)
* Deciding and extraction ROI points by plotting an image for both yellow and white mask
* Then apply hough transform for to detect lines from roads for both white and yellow mask
* Then save extracting lines for hough transform
* Calculate hough peak for both yellow and white mask
* Then we apply hough line into our main filtered image and plotting using extrapolation
* Calculate slop for left and right line and also make equation for extrapolate
* Then for turn detection I predict by looking at vanishing point using cross product from extrapolated lines
* Then plot everything on image like extrapolated lines, direction and lanes on each frame

**Result:**

* Then write these on video using our filtered images
* It has almost 250 to 300 frames (fps)
* It create a output video which show result
* And from step one close the image which we show on screen

**Features:**

* Lane detection and tell where you go i.e. right, left, straight
* Curve detection to avoid accident
* Fast processing within 5-10 seconds output video on screen
* 95% accuracy rate for curve and lane detection