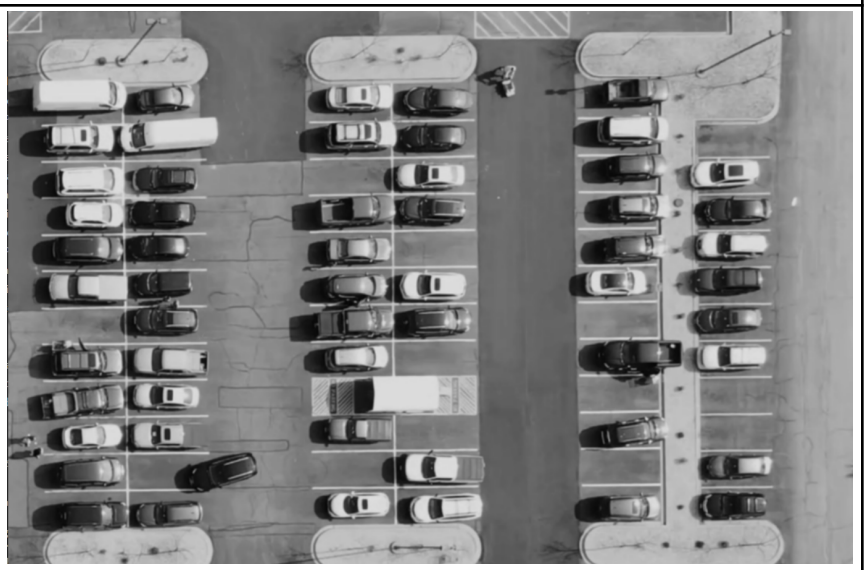
Project Development Phase
Model Performance Test

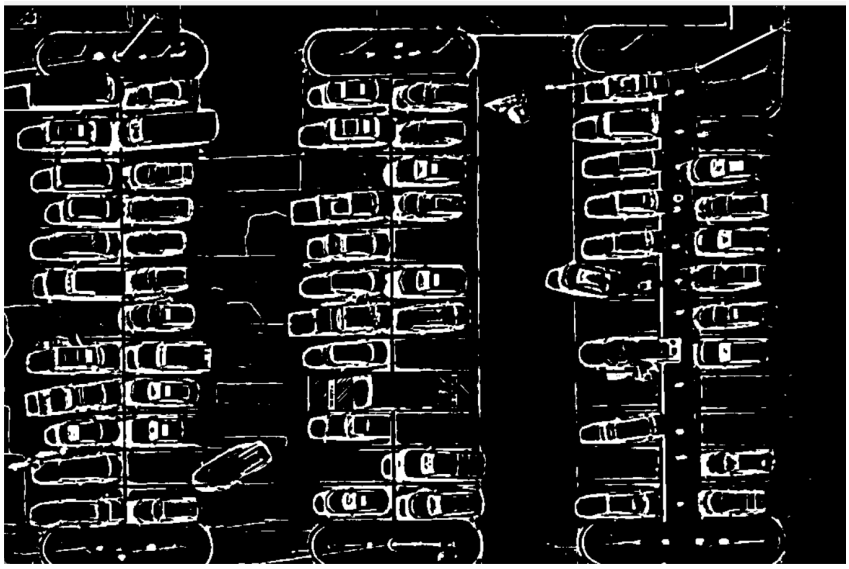
Date	15 November 2022
Team ID	Team-591796
Project Name	Project - AI Enable car parking using OpenCV
Maximum Marks	10 Marks

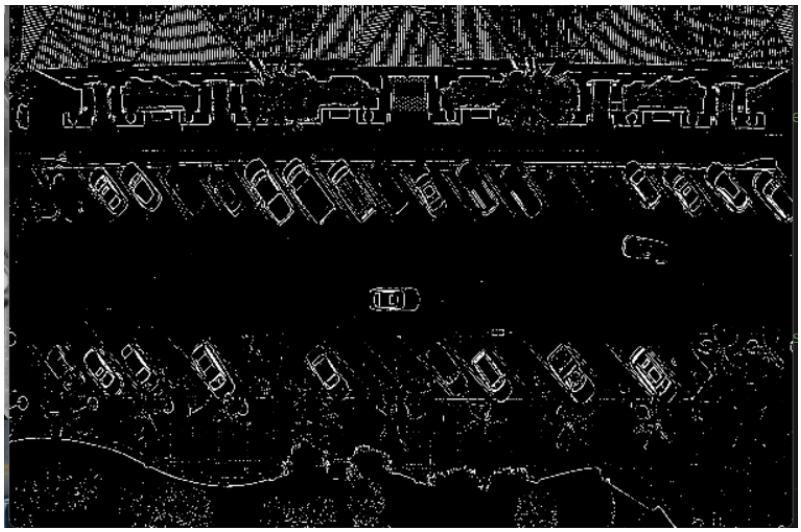
Model Performance Testing:

Project team shall fill the following information in the model performance testing template.

Model Summary: utilizes OpenCV to analyze a video feed from a surveillance camera in a car park. Its main objective is to detect and count available parking spaces based on predefined regions. The program starts by initializing the video feed and loading pre-defined parking space positions from a pickle file. For each frame, it applies various image processing techniques such as grayscale conversion, Gaussian blur, adaptive thresholding, median blur, and dilation to enhance the quality of the image. The checkParkingSpace function then iterates through the predefined parking positions, extracts the corresponding regions, and calculates the number of pixels to determine occupancy. The program displays the processed frame with visual indicators that differentiate between occupied and free parking spaces. This process continues in a loop until the 'Esc' key is pressed, providing a real-time user interface for monitoring parking space availability. The intermediate processing steps, such as blur and thresholding, play a crucial role in accurately identifying parking space occupancy and enabling accurate visualization and counting in a dynamic environment.

S.No.	Parameter	Values	Screenshot
1.	**Model Summary**	**Sample-1** Blur gray Image	

		Threshold image & Dilate image	
		Bounding boxes	

		Sample-2 Blur gray Image	
		Threshold image & Dilate image	
		Bounding boxes	

2.	**Accuracy**	**Sample-2**
		
		If count < 2170
		then it's a free space with green box
		else occupied space with red box
		`if count < 2170:`
		`  color = (0, 255, 0)`
		`  thickness = 5`
		`  spaceCounter += 1`
		`else:`
		`  color = (0, 0, 255)`
		`  thickness = 2`
		
		Sample-1
		
		count<750:
		then it's a free space with green box
		else occupied space with red box
		`if count<750:`
		`  color =(0,255,0)`
		`  thickness = 4`
		`  cvzone.putTextRect(img, str(count), (x,`
		`  spaceCounter+=1`
		`else:`
		`  color = (0,0,255)`
		`  thickness = 2`
		`cv2.rectangle(img, pos, (pos[0] + width, pos`