UIUC CS 445: Computational Photography

Final Project: Lane Departure Warning with CNN

Ryan Voge

**Motivation and Impact**

The project uses images from the point of view of a vehicle dashcam to predict whether a vehicle is traveling within the lane of a highway or departing from the lane. I am interested in understanding how images may be used to provide contextual information to support decisions we make in real life such as driving a vehicle. 360 degree cameras are now available on some in-market vehicles. This opens up many opportunities to use image information to improve the safety of people and property. In this use case, a lane departure warning could alert a drowsy or distracted driver to a potential deadly collision due to leaving the lane.

**Approach**

Images were collected from a phone mounted to a windshield for three scenarios (1) traveling within the lane, (2) departing lane to right, and (3) departing lane to left. A convolutional neural network was trained on the three classes of images and evaluated on a holdout sample for high accuracy. The resulting model prediction was used to score a new video with a message label for the frames where the lane departure was predicted.

**Results**

* Model Evaluation: The model produced an accuracy of 98.3% on the test dataset. Exhibit 3 displays the multi-class confusion matrix.
* Video Scoring
  + The model prediction was used to score a short video where the vehicle is being driven alternatively crosses over the left and right lane lines. The frames of the video are labeled with “Warning: Lane Departure!” where the model predicts lane departure.
  + The scored video may be viewed at the following link: <https://youtu.be/K2W1xaMGV3s>

**Implementation Details**

* Data Collection:
  + I mounted a smart phone to the top center of my windshield to collect input image data.
  + I chose a rural highway to collect the data due to minimal traffic. I needed to collect many images of departing the lane and wanted to minimize any potential risk to other vehicles.
  + Inlane class frames: I recorded video of driving with the lane consistently and then sampled frames from the video for the Inlane sample set. 842 frames were used for the Inlane class.
  + Depart Right and Depart Left classes frames: I took pictures while the vehicle crossed or was about to cross the line either on the right or the left. 166 and 175 frames were used for the Depart Right and Depart Left classes, respectively.
  + Exhibit 1 shows sample input images for each class.
* Model Dataset Creation:
  + Isolate image intensity: The frames were converted to HSV color space and only the V channel was used.
  + Align frame size: The input Inlane video frame size was 480 x 680 and the input Depart images frame size is 2160x2160. The frames were cropped or resized to 480 x 480.
  + Crop out sky and vehicle hood: For each frame, a patch of size 100 x 480 was extracted just above the vehicle hood for the model. More discussion on this is provided in the Challenges and Innovation section.
  + Exhibit 2 shows the sample input images for each class after the above adjustments
  + Training and Test datasets: The cropped, resized frames were combined into training and test datasets with 80% of each of the Inlane, Depart Right, and Depart Left frames being assigned to the training set. A label array was assigned for each frame with 1 for Inlane, 2 for Depart Right, and 3 for Depart Left.
* Model Training:
  + A convolutional neural network was used to train a multi-class classification model for the 3 classes. This model type was selected as it is designed for images.
  + This resource was used as a reference for CNN and as starter code: <https://towardsdatascience.com/convolutional-neural-networks-for-beginners-using-keras-and-tensorflow-2-c578f7b3bf25>
  + Several different model forms were tested. The final form has 3 convolution layers, with 2x2 max pooling layers, and a densely connected layer. The Relu activation was used in the activation layers and Softmax in the densely connected layer. The Adam optimizer was used.
* Languages: Python; Libraries: cv2, matplotlib, Keras, numpy

**Challenges and Innovation**

* **Challenge – Camera angle shifting and vehicle hood geometry**: The vehicle hood is in frame and contains some ridges and valleys. The data was collected over several vehicle trips, each time the phone was re-attached to the mount, the camera angle with respect to the hood was different. Since each vehicle trip focused on collecting either only Inlane, Depart Right, or Depart Left frames, the hood geometry in the frame became a significant indicator of the class. This resulted in very high accuracy in on holdout test sample because it was sampled from the same set as the training sample. However, it performed very poorly on the evaluation video because that was taken from yet another trip where the hood geometry was different. To remove the effect of the camera angle with respect to the vehicle hood, I cropped the frames to remove the hood from both the input data and the evaluation video frames for scoring the video. The differences in the hood perspectives can be observed in Exhibit 1.
* **Innovation – Scoring Video and Adding Warning Message**: The model evaluation on the test dataset demonstrated the high accuracy. However, scoring the video and providing a visual representation of the prediction is more powerful. To add the label, I created a .png image of the message. I scored all the frames of the video and when either the Depart Left or Depart Right Class was predicted, I added the message to the frame using a mask.

Exhibit 1: Sample Input Frames



Exhibit 2: Sample Processed Frames for Training Model (matches images in Exhibit 1)



Exhibit 3: Multi-Class Confusion Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Predicted Class | | | |
|  |  | InLane | Depart Right | Depart Left |  |
| Actual Class | InLane | 169 | 0 | 0 | 169 |
| Depart Right | 4 | 30 | 0 | 34 |
| Depart Left | 0 | 0 | 35 | 35 |
|  | 173 | 30 | 35 | 238 |