Intention Prediction of Pedestrians for Autonomous Vehicles

Jaimin Kachhadiya, Subham Swastik Samal
Virginia Tech
jaiminkachhadiya@vt.edu subhamsamal@vt.edu

Abstract

In deep learning-based autonomous driving applications, detecting and predicting the actions of pedestrians is a critical task, considering the safety issue involved. Deep learning models can be trained on large-scale datasets to recognize, and the models can learn complex features from the input data, such as the pedestrian's pose, surroundings, and trajectory. These features can then be used to predict the pedestrian's future actions and intent, enabling the autonomous vehicle to maneuver appropriately. However, developing accurate and robust deep-learning models for pedestrian action detection and prediction is still an active area of research and development in autonomous driving.

In this project, a framework was developed for predicting pedestrian crossing intentions in natural traffic scenes using video sequences obtained from an RGB camera mounted on an autonomous vehicle. The framework involves detecting and tracking pedestrians and predicting whether they will cross or not using an LSTM model trained on the JAAD dataset[8]. This project will contribute to the development of intelligent vehicles that can interact safely and efficiently with pedestrians in complex environments.

1. Introduction

To reduce traffic fatalities, an autonomous vehicle must be able to understand and anticipate the intentions of other road users. Compared to other road users, pedestrians are the most vulnerable when crossing the roadway because there aren't adequate protection measures in place. If an unmanned vehicle can recognize and anticipate people in a real-world driving situation like a human driver, as shown in figure 1., it will be able to take the appropriate actions and avoid potential collisions with pedestrians.

Thus, predicting pedestrian's behaviour and intent is one of the most critic capabilities autonomous vehicles should have. Previous approaches in this area include Markov Desision Processes[1], with its drwaback being the state at a particular time is only influenced by the input at the previous time, without considering earlier states. Dynamic Bayesian Networks have also been proposed in [6], but it requires accurate segmentation and efficient tracking of pedestrians, which is difficult.

In this project, a deep learning based framework is proposed to predict and comprehend the pedestrian's intention, consisting of the following components: detection, tracking, pose estimation and prediction. First, the pedestrian will be detected and tracked using YOLOv5 and DeepSort on the dataset. Then, the pedestrian's dynamical and contextual information would be extracted from the joint coordinates obtained using pose estimation algorithms like OpenPose and YOLOV7-pose model. Following this, the proposed model would predict the crossing intention of the pedestrian by an LSTM mechanism. Finally, in the experiment section, comparisons will be made between different LSTM Mechanisms.

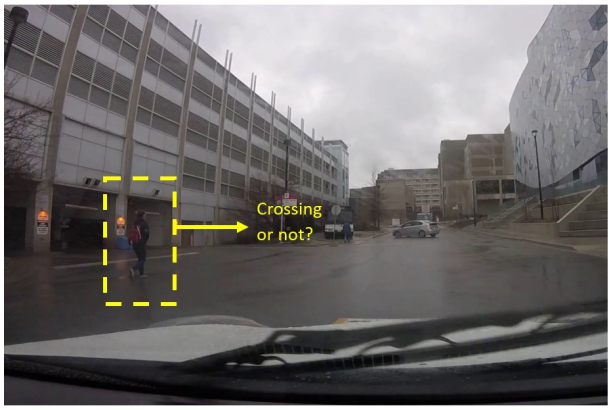

Figure 1: Our focus: Is the pedestrian going to cross?

2. Methods

In our proposed model, first, the model takes the video as an input and employs object detection and tracking algorithms to identify pedestrians and track each pedestrian's movement across the video frame. The results are the bounding box coordinated of the pedestrians. Furthermore, we obtain the coordinates of multiple human joints using

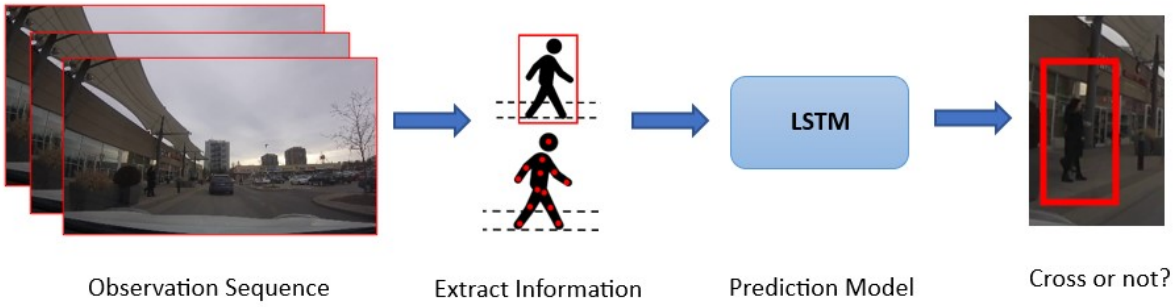

Figure 2: Schematic diagram of the overall framework

pose estimation algorithms for each frame of the video. Using the above methods, time-series data for relevant features was extracted for the video sequences. These were then trained by an LSTM architecture for learning the temporal relationships between the features and the actual label, to predict the pedestrian's intent to 'cross' or 'not cross'. Figure 2 describes the complete architecture of the proposed model.

2.1. Pedestrian Detection and Tracking

The YOLOv5[9][11] model is used in this study to identify pedestrians in video clips as it is one of the most recent single-stage object detection techniques, and better than other detection methods, including DPM and R-CNN. In the YOLO algorithm, an input image is divided into a S x S grid. A grid cell is in charge of detecting an object if its center falls within that grid cell. The bounding boxes and confidence scores for each box are predicted in each grid cell. These confidence scores reflect how confident the model is that the box contains an object and how accurate it thinks the predicted box is. For each grid cell, YOLO forecasts numerous bounding boxes. Based on whose prediction has the highest current IOU (Intersection over Union) with the ground truth, YOLO designates that predictor as being "responsible" for object prediction. The bounding box predictors get more specialized as a result of this.

The pedestrians are the only objects that are required to identify in our case as our proposed prediction model just needs the Bounding boxes information of the pedestrian and not any other objects in the traffic scene.

Once the pedestrians' bounding boxes have been identified in each image frame, it is necessary to implement the tracking of clearly identified pedestrians between frames using the unique object ID of each pedestrian. For this purpose, DeepSORT was chosen as the tracking algorithm. DeepSORT is a recent algorithm for tracking that extends Simple Online and Real-time Tracking[2] and has shown remarkable results in problems involving multiple object tracking. It takes advantage of pedestrians' visual data to improve tracking efficiency. DeepSORT enables the generation of pedestrian re-identification features that can be compared with pedestrian appearance within the obtained Bounding Boxes output by the detection module to minimize the number of identity switches.

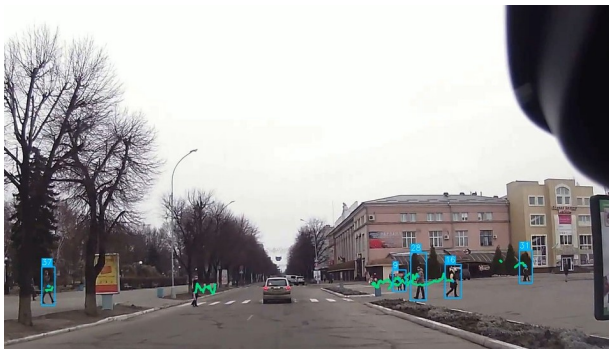

Figure 3: Pedestrian detection and tracking: The bounding boxes at the current timestep marked with the pedestrian ID for tracking, the recorded trajectories are noted in green.

2.2. Pose Estimation

Two well-known architectures, OpenPose[4] and YOLO V7 Pose were implemented to attain estimates of multiple 2D human joint coordinates across all frames of the videos. Openpose extracts 18 2D landmarks on the human body, while YOLO V7 extracts 17. Both of the architectures were tried out, and YOLO V7 Pose was found more suitable for our purpose. The YOLO V7 Pose was significantly better when certain body parts are occluded; and performed better in lower lighting conditions. After obtaining the key points, several features were computed, like distances be-

tween points, relative angles between pairs of keypoints etc., which convey the motion dynamics.

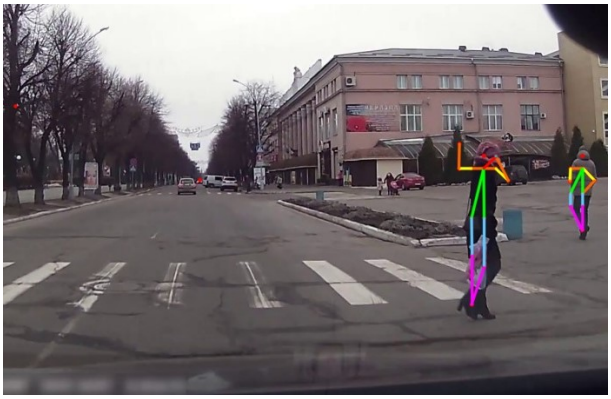

Figure 4: Pose Detection: Obtaining joint coordinates

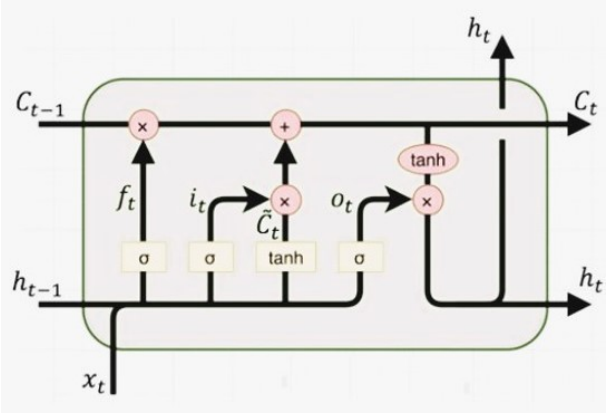

Figure 5: Structure of LSTM cell

2.3. LSTM based intention prediction

RNN-based approaches have produced reliable results in sequence prediction challenges, as shown in [5][12][14], thanks to their capacity to describe intricate temporal correlations in the input sequence data they receive. The typical RNN is unable to make reliable predictions when memorizing past lengthy sequences in practical applications, hence LSTM was chosen as the prediction sub-fundamental model's model.

An LSTM's memory block is made up of one or more memory cells, and as shown in Fig. 5, each memory cell shares three gates (input, output, and forget gates) that control and protect the cell state. The input gate is in charge of managing input data contributions to the updating of the memory state. How much of which of the previous state contributes to the current state of the memory block is determined by the forget gate. Based on input and the block's memory, the output gate chooses what to transfer to the hidden state.

The memory cells in LSTM memory cells are calculated and updated each time step t basis the following Equations:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\bar{C}_t = tanh(W_C[h_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * tanh(C_t) \quad (6)$$

where i_t, f_t, o_t and C_t represent the actions for the input gate, forget gate, output gate, and cell state updates at time t respectively. σ represents the sigmoid function, x_t is the input at the current timestamp, and h_{t-1} and h_t are the outputs at previous and current timesteps respectively. W_k and b_k are the weights and biases for the respective gates(k). The operations of various gating units based on the operations marked by Eqs 1-6 enables the LSTM to obtain the temporal properties of the input information when the sequence information is fed into the LSTM.

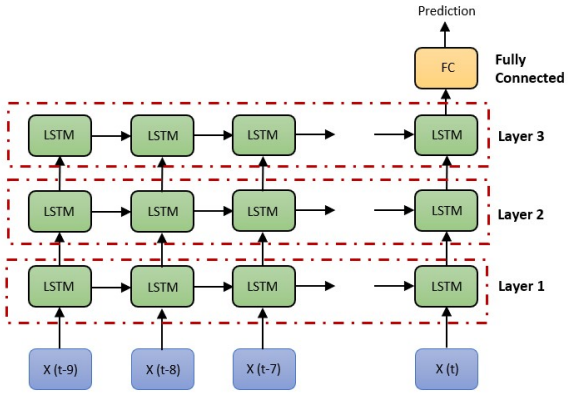

Figure 6: The proposed LSTM model for pedestrians's intent prediction

In our proposed approach, a deep network of stacked LSTM blocks was built for the task of intent prediction. As shown in figure 6, the stacked LSTM network consists of multiple stacked LSTM layers. Firstly, an input sequence of a window size of 10 of extracted features as discussed in Sections 2.1 and 2.2. The input layer gives its input to the first layer of our stacked LSTM network, where the number of neurons is equal to the input sequence window size. The first LSTM layer in turn feeds into the second LSTM layer, which afterwards feeds into the final LSTM layer. As

Figure 7: Visualization of the result: The images are the 10th image in their respective sequence of 15 images (Sec 3.3), C or NC refer to predictions Crossing or Not Crossing). Here C or NC is determined by whichever occurs maximum times in the predictions of the next 5 steps.

a result of the hierarchically stacked representation layers, the final learnt model is better able to understand the temporal dependency of its input sequence, which leads to a more accurate prediction. In these 2 layers, different number of neurons are tried out in our experiments Finally, the last LSTM layer connects into a fully connected layer in which the number of neurons is the number of timesteps ahead for which the prediction is to be done, which is taken to be 5 in our experiments. The activation function for this layer taken to be the sigmoid function to obtain the probability of the pedestrian's crossing intent. Thus, to perform this, our videos were divided into sets of 15 consecutive frames, with the first 10 used to train the prediction intent for the next 5 timesteps.

2.4. Dataset

PIE dataset [7](Pedestrian Intention Estimation) and JAAD dataset [8] (Joint Attention for Autonomous Driving) are two commonly used datasets for pedestrian intention prediction using LSTM (Long Short-Term Memory) models.

The PIE dataset is a large-scale dataset that contains videos of pedestrians in various urban scenarios. It includes total of 6 hours of video split into 36 video clips of pedestrians taken from different camera angles and distances. Each video clip is approximately 10 minutes long. The JAAD dataset, on the other hand, contains over 346 video clips of pedestrians recorded from a forward-facing camera on a car. The video sequences in the JAAD dataset are shorter, about 5 $\sim$ 15 seconds in length, and include annotations for pedestrian intention and vehicle-pedestrian interactions. This JAAD dataset was collected in naturalistic driving scenarios in various lighting conditions, weather conditions (rainy, windy, snowy etc.) across multiple countries [8], whereas the PIE dataset was entirely recorded in downtown Toronto, Canada during daytime under sunny/overcast weather conditions[7].

While the PIE dataset is a comprehensive dataset with a long video clips, using the JAAD dataset may be more suitable for pedestrian intention prediction using LSTM models due to its shorter video sequences. This is because LSTMs are designed to handle sequences of variable length, but longer sequences can result in computational challenges and memory constraints. By using shorter video sequences, the JAAD dataset can be more efficient for training and testing LSTM models. For our experiments, the dataset is divided into three sections: 70% for training, 10% for validation, and 20% for testing.

3. Experiments and Discussions

Different LSTM model architectures were tried by modifying the number of neurons in the 2 hidden layers, dropout rates, and optimizers. A batch size of 512 training samples were used. The loss function was considered to be binary cross entropy and the learning rate was considered to be 0.001. Early stopping was enabled with a patience of 5 epochs on validation loss to stop overfitting. For evaluation, the accuracy metric is considered. Table 1 represents the scores of the different model architectures.

It was observed that models with higher number of neurons (128,64) performed better compared to lower number of neurons (64, 32). Adam optimizer performed better for the networks with higher number of neurons, whereas the performance was almost similar for (64,32) layers. Very

Model	Hidden neurons	Dropout(%)	Optimizer	Train Acc(%)	Val Acc(%)	Test Acc(%)
M1	64,32	0	Adam	60.3	58.9	58.5
M2	64,32	0	SGD	61.4	59.2	59.4
M3	64,32	20	Adam	57.5	56.6	57.0
M4	64,32	20	SGD	58.2	55.4	55.6
M5	128,64	20	Adam	74.5	71.1	70.7
M6	128,64	20	SGD	69.4	66.7	68.2
M7	128,64	0	Adam	70.1	67.7	68.2
M8	128,64	0	SGD	68.4	69.1	66.7
M9	64,32	50	Adam	51.2	45.7	44.8
M10	128,64	50	Adam	54.2	47.6	46.1

Performance comparison for different architectures

Observation Length	5	6	7	8	9	10	11	12	13
Test Accuracy(%)	60.4	65.7	68.3	67.7	69.0	70.7	70.2	68.7	69.5

Table 1: Performance comparison for different observation lengths

high dropout rate led to a severe decrease in performance, wheres the performance for 0 and 20 % dropout was almost similar.

The highest accuracy was obtained for the LSTM model with 128 and 64 neurons, considering a 20% dropout rate and using Adam Optimizer. A visualization of one of the results is presented in Fig 7.

The impact of input observation length on the prediction accuracy was also observed. The model with best performance in Table 1, i.e. M5 was considered for this experiment. Generally, longer input sequences imply more valuable information for prediction. Thus, as shown in Table 2, as the length of the observation sequence increases, generally the accuracy of LSTM improved, while as the observation sequence length was increased to beyond 10, the accuracy stayed more or less the same. This could possibly be because with longer input sequences, the introduced invalid and noisy information also gets accumulated, countering the more valuable information obtained with longer sequences.

4. Conclusion

In this project, the crossing intent prediction of pedestrians was performed using LSTM by considering the pedestrians' positions and pose features. The highest accuracy of 70.7% was achieved. Our accuracy is not very high compared to recent research done on this dataset. One of the key factors could be the absence of contextual features, i.e. the features surrounding the pedestrians, which includes other features such as the number of other vehicles, traffic lights, zebra crossings etc. Some research which have considered these features have achieved more than 85% accuracies [13][3]. Further, in our method we are performing predictions for the next 5 timesteps using the 10 timesteps data. In the future, we would try to do the prediction for only 1 timestep during the training, and then recursively forecast the variables for the future timesteps in the prediction window as done in [10].

5. Contributions

The contribution of this project involves the collaboration of Subham and Jaimin, who worked on different aspects of the pedestrian intention prediction model. Subham contributed to the implementation of the YOLO and DeepSORT tracking algorithm, which involved leveraging the visual data of pedestrians to improve tracking efficiency. On the other hand, Jaimin worked on the pose estimation using the YOLOv7 pose model, which allowed for the extraction of relevant features for the LSTM models. Both Subham and Jaimin trained the LSTM models with different configurations, studied their performances, and compared the results. Finally, the trained models were tested on the test video sequences, and the predictions were evaluated. The collaborative efforts of both team members resulted in the successful implementation of a pedestrian intention prediction model, which has the potential to improve the safety of autonomous vehicles on the road.

References

[1] T. Bandyopadhyay, C. Z. Jie, D. Hsu, M. H. Ang, D. Rus, and E. Frazzoli. Intention-aware pedestrian avoidance. In *Ex-*

perimental robotics: The 13th international symposium on experimental robotics, pages 963–977. Springer, 2013. 1

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016. 2

[3] S. A. Bouhsain, S. Saadatnejad, and A. Alahi. Pedestrian intention prediction: A multi-task perspective. *arXiv preprint arXiv:2010.10270*, 2020. 5

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 2

[5] B. Dong, H. Liu, Y. Bai, J. Lin, Z. Xu, X. Xu, and Q. Kong. Multi-modal trajectory prediction for autonomous driving with semantic map and dynamic graph attention network. *arXiv preprint arXiv:2103.16273*, 2021. 3

[6] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila. Context-based pedestrian path prediction. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 618–633. Springer, 2014. 1

[7] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6262–6271, 2019. 4

[8] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 206–213, 2017. 1, 4

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[10] K. Saleh, M. Hossny, and S. Nahavandi. Intent prediction of pedestrians via motion trajectories using stacked recurrent neural networks. *IEEE Transactions on Intelligent Vehicles*, 3(4):414–424, 2018. 5

[11] M. Sukkar, D. Kumar, and J. Sindha. Real-time pedestrians detection by yolov5. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 01–06. IEEE, 2021. 2

[12] J. Sun, Q. Jiang, and C. Lu. Recursive social behavior graph for trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 660–669, 2020. 3

[13] D. Yang, H. Zhang, E. Yurtsever, K. A. Redmill, and Ü. Özgüner. Predicting pedestrian crossing intention with feature fusion and spatio-temporal attention. *IEEE Transactions on Intelligent Vehicles*, 7(2):221–230, 2022. 5

[14] W. Zeng, Z. Quan, Z. Zhao, C. Xie, and X. Lu. A deep learning approach for aircraft trajectory prediction in terminal airspace. *IEEE Access*, 8:151250–151266, 2020. 3