# Pedestrian Intention Prediction

Haziq Razali, Alexandre Alahi
Visual Intelligence for Transportation Laboratory, EPFL

**Abstract**

The ability for an autonomous vehicle to predict and anticipate human action is a critical prerequisite for operating in urban areas. Annotated datasets to train machines for such tasks are however, in short supply and are very expensive to build. In this paper, we introduce a dataset of recordings at non signalized crossings for the study of pedestrian behaviour. The key feature of our dataset is that it is recorded from a static camera and thus opens the possibility for it to be automatically annotated. We also then experiment with recurrent neural networks to determine the limitations of such a dataset. This work is available at `https://github.com/HaziqRazali/Pedestrian-Intention-Prediction`.

*Keywords:* , Autonomous Vehicles, Pedestrian Intention Prediction, Human Behaviour Analysis

## 1. Introduction

Pedestrians exhibiting the intention to cross are normally identifiable through a certain set of cues. For instance, they turn their heads to look for incoming traffic as they approach the crosswalk. They also do not cross and instead wait at the sidewalk if there are nearby vehicles that have not come to a stop. The ability to learn these behaviours and use them to predict human motion in urban areas is extremely valuable for the deployment of autonomous vehicles. A growing trend at tackling such learning problems is to utilize a framework that is driven by deep neural networks. One caveat of this is however, its requirement for large amounts of training data when looking to solve a completely new challenge. In the context of autonomous vehicles, there is not much egocentric datasets aimed to analyze pedestrian behaviour at road crossings [1, 2], the collection and annotation of which would require many man-hours. Thus in this paper, we experiment the use of static cameras for the study of pedestrian intention at road crossings. Our hypothesis is that the behaviour of a pedestrian when observed by a static camera should be no different than if he or she were observed by a moving one (mounted on a vehicle). The main advantage of using static cameras as opposed to moving ones is that it enables us to automate the annotation process, making it significantly easier to both maintain and expand the dataset. As such, our contribution in this paper is twofold:

- We introduce a dataset of static recordings of non-signalized road crossings that enables behavioral analysis of pedestrians at the point of crossing.
- We study the feasibility of using a deep neural network that has been trained on recordings from a fixed camera, on egocentric non-static recordings, then investigate improvements that can be made.

The remainder of this paper is organized as follows. We begin in section 2 by introducing our dataset. We then describe our experimental setup in section 3 and present the results of our experiments in section 3.3. Lastly, we conclude and discuss improvements in section 4.

## 2. Dataset

### 2.1 Data collection

The data is collected in Lausanne using the GoPro Hero 6 Black. The videos are recorded at 1920x1440 at a frame rate of 30 fps. Figure 1 shows a sample frame of the videos with the duration of the recordings.

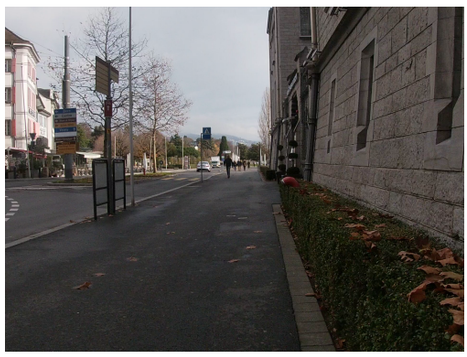
### 2.2 Ground truth

We built the ground truth by running the Mask R-CNN [3] object detector followed by a Hungarian tracker over a region of interest in the image. Since the aim is to predict the pedestrian's decision before the point of crossing, we only need to track the pedestrian until he either enters the crossing (to cross to the other side of the road) or until he leaves the frame. As such we set the said region of interest to only cover one side of the sidewalk and a small portion of the road. The label of

(a) Ouchy-1, 1hr

(b) Ouchy-2, 2hr30min

(c) Ouchy-3, 1hr



(d) Ouchy-4, 1hr

(e) Olympic-1, 45min

(f) Olympic-2, 45min

(g) Lausanne-Gare-1, 30min

(h) Lausanne-Gare-2, 30min

(i) Riponne-1, 15min

(j) Riponne-2, 15min

Figure 1: Sample frame of the videos

each pedestrian is then determined by checking if his tracks end up in the road. Lastly, noise in the ground truth is minimized by removing tracks whose lifetime is below some threshold. The statistics of the ground truth given the current framework are summarized in 1. In the table, the entry Ouchy-1-Left implies that the ground truth was generated by running the detector and tracker on the left portion of the Ouchy-1 video (refer to figure 1) containing the sidewalk and a small area of the road. Recordings that have not been processed to generate the ground truth are not in the table. We also mostly run the detector and tracker on the side of the image that is less obstructed by vehicles to reduce errors, a limitation of the current framework.

| **Video** | **Duration (hr:min)** | **# Crossed** | **# Did not cross** |
|---|---|---|---|
| Ouchy-1-Left-and-Right | 1:00 | 214 | 402 |
| Ouchy-2-Left | 2:30 | 122 | 666 |
| Ouchy-3-Right | 1:00 | 41 | 121 |
| Ouchy-4-Right | 1:00 | 51 | 167 |
| Olympic-1-Left | 0:45 | 22 | 33 |
| Olympic-2-Right | 0:30 | 31 | 71 |
| Riponne-1-Left | 0:30 | 15 | 74 |
| Lausanne-Gare-2-Right | 0:45 | 31 | 101 |

Table 1: Statistics for some of the recordings.

# 3. Experimental setup

## 3.1 Architecture

Our baseline is depicted in figure 2. The architecture works by passing at each timestep, an image crop of the pedestrian to the CNN feature extractor to produce a high dimensional feature vector. That vector is then passed into the recurrent network for sequence learning. A fully connected layer followed by the sigmoid function at the final timestep outputs the label probabilities. Intuitively, the architecture should learn to detect that changes in the orientation of the pedestrian's head and/or body is indicative of his intention to cross. In our implementation, we use the output of the last convolutional layer of the VGG-16 [4] pretrained model as input to the LSTM with input, hidden, and output sizes of 32,16,16 neurons respectively. We tried using Resnets [5] in our experiments but found that it provided almost no improvement but at the cost of a noticeable increase in the training time. As such, we decided to stick to the VGG-16 network.

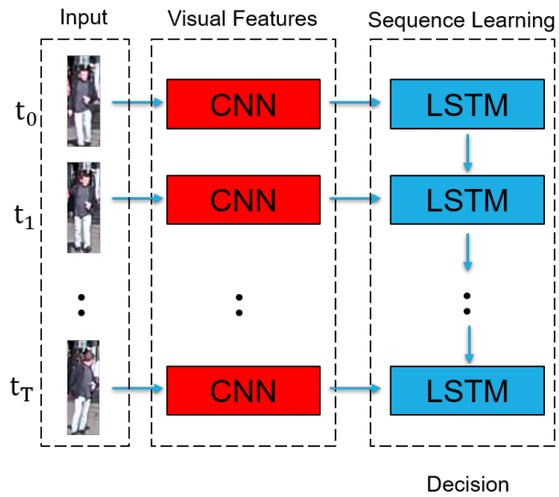



Figure 2: A standard CNN LSTM.

## 3.2 Training details

We initialize all non pre-trained parameters using the He initialization of [6] and use a one-hot encoding of the labels. We optimize over the binary cross entropy loss using stochastic gradient descent with a batch size of 16 and use the learning scheme of RMSProp with a base learning rate of 0.0005, decay of 0.9 and with no momentum. We unroll the LSTM for 8 time-steps and train the model for 200 epochs. The models were developed in PyTorch.

We train the architecture on the final 8 crops of each pedestrian, spaced 15 frames apart. For pedestrians that crossed the road, we use the crops that are not in the crossing. We resize the

images to 140x80 and augment the dataset by using 100x60 crops and horizontal flips. Lastly, no pre-processing was applied to the images besides linearly scaling to the range of the sigmoid activation function [0, 1].

### 3.3 Experiments

*3.3.1.* Results

In the first experiment, we evaluate the performance of the trained classifier when trained on our dataset. We trained it on a portion of the Ouchy-1-Left sequence and tested it on the remaining portion of the Ouchy-1-Left sequence, the Riponne-1-Left and Lausanne-Gare-1-Right sequences as well as JAAD dataset [2]. We did not train nor test it on the sequences which we feel contained a lot of errors in the ground truth. The results are summarized in table 2. It can be seen that the classifier performs quite well on the Ouchy-1-Left sequence but quite badly on the rest. These findings probably suggest that the classifier is unable to generalize over different scenes where the resolution of the pedestrian might be different due to the distance of the camera from the crossing or just changes in lighting. Overall results on the JAAD dataset is also quite poor, most probably due to the fact that the dataset contains pedestrians walking in all directions, a feature that is not observed in our dataset. Lastly, we would like to point out that the numbers on our dataset may not represent the true performance of the classifier the due to errors in the ground truth induced by the tracker.

| **Set** | **Video** | **# Positive/Negative Samples** | **Precision** | **Recall** |
|---|---|---|---|---|
| Train | Ouchy-1-Left | 84/501 | 0.96 | 0.94 |
| Test | Ouchy-1-Left | 38/165 | 0.85 | 0.74 |
| | Riponne-1-Left | 15/74 | 0.39 | 0.60 |
| | Lausanne-Gare-1-Right | 31/101 | 0.31 | 0.51 |
| | JAAD | 125/128 | 0.53 | 0.14 |

Table 2: Results of the baseline.

In the second experiment, we trained and tested the model on the JAAD dataset to view its performance when taking in sequences that have been recorded from a moving camera. The dataset consists of 346 5-10 second recordings from a moving camera, of which 300 was used in the training set and the remaining 46 used in the test set. Additionally, the dataset contains labels that describe the state the pedestrian is currently in i.e. whether or not he is standing, walking, looking at the camera, or crossing etc. The label 'crossing' denotes the instant the pedestrian begins to cross the road, the exact moment of which has been manually inferred by the authors. The design of the architecture follows that described in section 3.1 with the addition of one linear classifier for each of the actions, standing, walking, looking and hand-waving at every timestep. The training scheme follows that described in section 3.2 except that we take the subsequence before the decision to cross instead of the taking the subsequence before the zebra crossing. The results of this experiment are presented in table 3. The results show signs of overfitting as the training statistics are way better than the test ones. It also was noted during training that the validation loss curves always increase after the first few iterations. However, we believe that this is a promising start as the precision score of 0.5 was attained with a much smaller number of samples at 200 but is in between the two baselines developed by [2] at precision scores of 0.39 and 0.62 that was attained with 3324 samples in their experiments. Lastly, we tried excluding a combination of the above-mentioned actions but to no avail.

| **Set** | **Video** | **# Positive/Negative Samples** | **Precision** | **Recall** |
|---|---|---|---|---|
| Train | JAAD (1 - 300) | 106/94 | 0.81 | 0.91 |
| Test | JAAD (301 - 346) | 19/34 | 0.50 | 0.79 |

Table 3: Accuracies of the baseline. The model was trained on the sequence Ouchy-1 and tested on the other two.

*3.3.2.* Guided Backpropagation

In our third experiment, we visualize pixels that positively affect the output class through a method called guided back-propagation [7]. This technique of analysis works by computing the derivative of the class score with respect to the input, but back-propagates through ReLU layers the quantity,

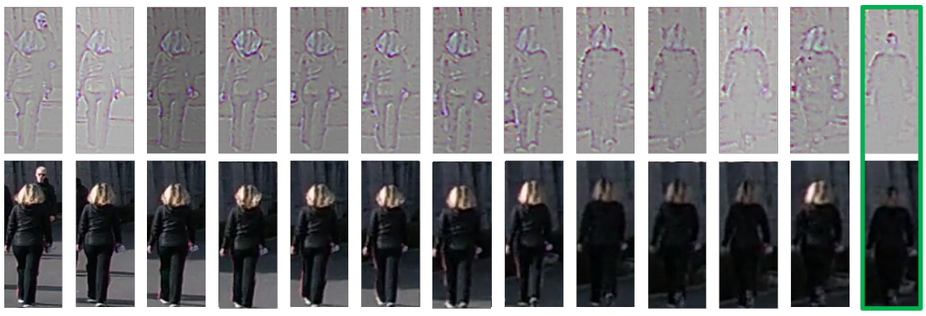
$$\mathbf{1}_{\{s>0\}}\mathbf{1}_{\{\frac{\partial l}{\partial x}>0\}}\frac{\partial l}{\partial x} \tag{1}$$

thus keeping only units which have a positive contribution to the final neuron. The results of this experiment are displayed in figure 3. In the images, the pixels that are not gray in color are those that are deemed important by the architecture with its importance emphasized by the

intensity. Also, note that the entire sequence with exception of the final timestep (image) has a classification output of "not-crossing". The output is guided back-propped when its label turns from "not crossing" to "crossing" as denoted by the red bounding box. For pedestrians that did not cross the road, we simply back-prop at the final timestep, denoted by the bounding box in green.

On first inspection, it can be observed that the architecture is using information pertaining to the pedestrian due to the silhouettes that can be seen on the image. This might suggest that the architecture is making its prediction based on the orientation of the pedestrian's body and/or head. Secondly, it can be observed that the architecture is also using background information as the pixels that do not constitute the pedestrian are not of a flat gray in color. This shows that the model is incorrect in its assessment as the background content of the pedestrian crop should not provide any information regarding his decision to cross or not. This is probably due to the lack of variety in the training samples, even with data augmentation. A solution to this issue would be to simply mask the background so as to force the architecture to look only at the pedestrian when making its decision. Lastly, it can also be observed that all the images are somewhat similar in intensity, indicating that the architecture is not looking at specific frames when making its decision. Intuitively, the frames that contain the pedestrian in a different orientation should be the ones that are most significant when trying to determine his intention to cross. This is probably due to the additive feature of the cells that prevent the vanilla lstm from focusing at specific timesteps and thus suggests the need for an attentive module or an entirely different architecture. On a final pessimistic note, it might be possible that the results presented here are all not conclusive due to the previously mentioned lack of variety in the training samples.

(a)



(b)

Figure 3: Visual results of the guided backpropagation algorithm

## 4. Conclusion

We have presented a dataset for the prediction of pedestrian intention at road crossings. The key feature of our dataset is that it allows users to automate the annotation process, making it easy to expand the dataset. Experimental results show that the CNN-LSTM is able to detect that changes in a pedestrian's orientation is indicative of his intention to cross but that the current dataset is not sufficient for the model to generalize well.

We believe that there are several improvements that can be made which might further enhance the performance of the model developed in this paper. Firstly, the use of a pose estimation network could prove useful as (1) using keypoints as input to the LSTM would be a better representation of the pedestrian's orientation as opposed to an output from a CNN in addition to the fact that (2) the size of the feature vector representing the keypoints would be significantly smaller than the size of the feature vector coming from the VGG16, thereby reducing the complexity of the architecture and consequently, the likelihood of overfitting. Also, better data augmentation techniques to both increase and further diversify the training samples would surely work well.

## References


[1] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive bayesian filters: A comparative study," *International Conference on Computer Vision*, 2017.

[2] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," *International Conference on Computer Vision*, 2017.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *ArXiv*, 2017.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv*, 2014.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *ArXiv*, 2015.

[6] ——, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *ArXiv*, 2014.

[7] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *ArXiv*, 2014.