Pedestrian Intention Prediction

Technical report

CIVIL-459

Authors

Arina Rak

Chang Chun-Tzu

June 13, 2023

1 Initial approach

In this work we strove to adapt the paper Pedestrian Stop and Go Forecasting with Hybrid Feature Fusion [1] which was used on Stop / Go transition prediction to Intention prediction. We were working from the authors' implementation, modifying it to our needs.

1.1 Data Adaptation

Since we were changing the task within the JAAD dataset [2], the first thing we needed to do is adapt is data processing. In that, we a) removed attributes such as action ("walking" / "standing") and pedestrian motion direction, to avoid data leak. We used the "crossing" / "non-crossing" state of the pedestrian in prediction-horizon frames (1-2s) as the label and were taking a history sequence of frames (1-2s) to make the prediction.

Apart from that, we perform class balancing (by undersampling) for the train and validation sets, to expose the model equally to both classes.

With these modifications, we were able to train the system on our data. We used overfitting on the batch as a sanity check throughout our development process as well as monitoring the train and validation losses as well as running quality metrics.

1.2 Initial results

We got a **0.812** f1 score on the test set as our initial result, which seemed promising. However, we needed a reference point to compare our results. For that, we used majority voting (always predicting crossing, since the provided JAAD test split is highly imbalanced), which gave us an f1 score of **0.81**. The second simple baseline we used was outputting the label for the current frame as a prediction in prediction horizon frames. Two points should be mentioned about this baseline though: a) it only focuses on the simpler cases, where there are no changes in the pedestrian's behaviours, while we would like the model to focus on the transitions; b) usually a baseline is used to evaluate the "helpfulness" of the introduced approach, i.e. does it make sense to train such model/system or a simpler approach is already capable to do the job. Though this simple baseline allows us to get a reference point for our quality metric, it can not be used in real-life systems, since the labels for the current frame are unavailable, unless extracted with a separate

model. With this baseline, we acquired an f1 score of **0.84**.

2 Issues

During this preliminary study, we realized that our model is lagging behind and were looking for approaches to improve our performance. We performed a wide range grid search to find the best set of hyper-parameters, played around with prediction horizon and history length, as well as applied data augmentations to the images. However, all of those modifications only led to marginal improvements. In order to identify the source of the problem in the hybrid model, we decided to employ an ablation study. This involved systematically removing one of the four modules to observe its impact. This approach was taken due to the consistent underperformance of the hybrid model compared to the simpler baseline model.

2.1 Ablation

Following the experiments, a noteworthy discovery was made in Table 1: removing the scene description resulted in a significant performance decrease (the model stopped predicting crossing). This prompted us to investigate the contents of the scene description more closely.

Those include $S = (s_{tl}, s_{in}, s_{de}, s_{si}, s_{td})$ — binary / count-based scene characteristics

- s_{tl} — number of traffic lanes
- s_{in} — if is an intersection
- s_{de} — if an intersection is designated with a zebra crossing or a traffic signal
- s_{si} — if an intersection is signalized
- s_{td} — traffic direction (one/two-way)

We hypothesize (and this matches a small study we did on the test set) that in the JAAD dataset most of the videos where pedestrians are crossing the road include a road intersection or a crosswalk, so the model can easily use only those features to predict the "crossing" label. Another reason is that the scene features are concatenated to the outputs

of all RNN encoders (Figure 1), so it only goes through one linear layer which receives the strongest gradient updates (without backprop through time or gradient saturation due to multiplications in the chain rule). Therefore, this feature is leaking the test data and making the model find a shortcut, instead of performing actual training. Subsequently, we decided evaluting the model solely using the scene description and were surprised to discover that the performance results were comparable to those achieved by the entire hybrid model. This observation led us to suspect that the model had become overfit to the scene description, relying solely on it to make decisions.

Table 1: Ablation study results

	hybrid	**no CNN**	**no pv(bbox)**	**no behavior**	**no scnene**	**scene only**
f1	0.79	0.8	0.84	0.72	0.00	0.83

We assume, that the model is only relying on the scene and the rest of the features are actually "hurting" it to make the prediction. That is why we see increased performance in some cases of the ablation.

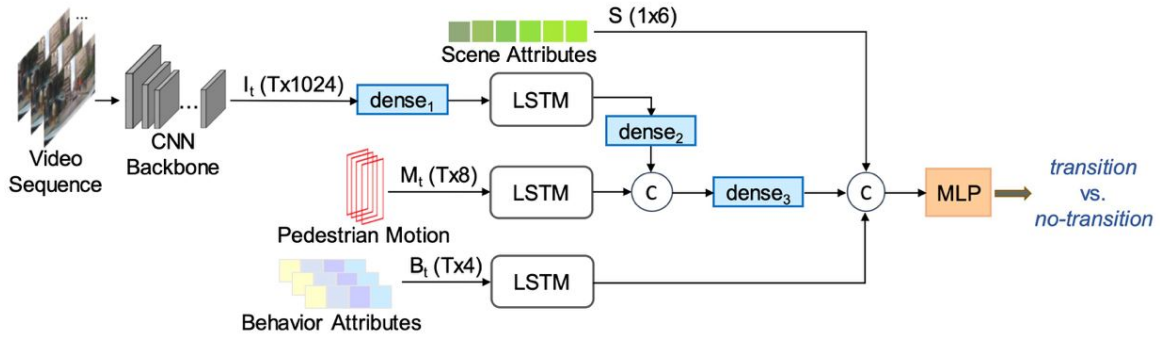

Figure 1: The Pedestrian Stop and Go Forecasting with Hybrid Feature Fusion

2.2 Experiments

We performed a set of experiments to try and locate the problem. For this set of experiments, we were always using the prediction horizon of 2 seconds and a history of 2 seconds, where applicable. We will provide the experiments in detail in the following:

Experiment 1: Training without scene description

The first step we took was training the whole system without the scene description, since it was the most powerful feature, preventing the rest of the system from learning.

However, with this setup the model was unable to distinguish between crossing and non-crossing, and selecting the optimal threshold for prediction (based on the validation set f1) always lead to the model outputting only the crossing class.

With this in mind, we decided that our visual encoder (ResNet18, pretrained on ImageNet) was experiencing too much of a domain shift and was not able to extract relevant information for the crossing / non-crossing task. To adapt it to the task we decided to try and fine-tune it on our downstream task first and then plug it into the system.

Experiment 2: Training the visual encoder

The first issue that we encountered while trying to train the visual encoder was that we couldn't successfully pass the batch-overfit sanity check.

The issue was due to the mismatch of the model's outputs in .train() and .eval() modes. We hypothesise that this is due to the mismatch of the feature distributions in the ImageNet and JAAD images and therefore the mismatch of batch norm statistics, we were able to overcome those issues by turning off the tracking of running statistics.

Unfortunately, when training ResNet18 on the whole dataset we also experienced an overfit: the training loss was decreasing, while the validation loss was increasing. Even with that, we were able to get better results than the whole system without the scene: **0.79** test f1 score.

Experiment 3: Whole image

Another direction of experimentation was varying the image input. In the stop/go prediction a crop of an image focusing on the pedestrian was passed to the visual encoder. However, for our task (especially once we get rid of the scene description) only pedestrian-crop might not be sufficient. What we tried next was providing the model with the whole frame instead (halved in size in each dimension to fit our GPU memory constraints).

With this the whole system started training, however, the results were just on par with always predicting the current frame's label baseline (**86.56** vs **86.57** test f1 score).

Experiment 4: Wider cropping of pedestrian

Instead of using the whole image and resizing it, we decided to modify the cropping approach to include only the wider area align with the height of the pedestrian bounding box in the input image. This adjustment ensured that the CNN module now focuses on not only the pedestrian but also the relevant environmental information. We hypothesized

that this change would yield improved results since the CNN module would then possess more powerful and comprehensive information.

In this experiment, we got comparable results than with using the whole image for training the ResNet18, while gaining a speed up in training time. That's why we default for this setting for the rest of the work.

Experiment 5: Fighting Visual Encoder Overfit

With the CNN-only experiments, we were still experiencing the overfit (validation loss kept going up), so we were progressively freezing more and more blocks of the architecture to reduce its capacity, apart from that we tried to insert a dropout layer into the final classification head and played around with it's values as well as L2 weight normalization. Training any number of layers on the CNN backbone led to overfit, so we decided to only train the final visual feature embedding layer and to reuse it in a whole system.

3 Final solution: bottom-up model

After carefully considering all the factors, we couldn't isolate the problem in the system as a whole and decided to build it in a bottom-up fashion, with the ability to analyze the contribution of each component. In line with this approach, we made the decision to exclude scene descriptions and behaviour data from the model. The exclusion of scene descriptions was motivated by the desire to avoid overfitting on specific scene attributes and help model to concentrate more effectively on learning other pertinent information, as mentioned above in experiments. Similarly, we opted to remove behavior data from the system due to the challenges associated with obtaining accurate behavior data in real-life scenarios, ensuring that our system remains applicable and practical in real-world applications.

Before training the hybrid model, we separately trained the CNN encoder with a ResNet18 backbone (image module, see Figure 2) and the LSTM encoder (pedestrian motion, see Figure 3). These individually trained models were then utilized as pretrained checkpoints during the training of the final hybrid model, facilitating a more effective learning process.

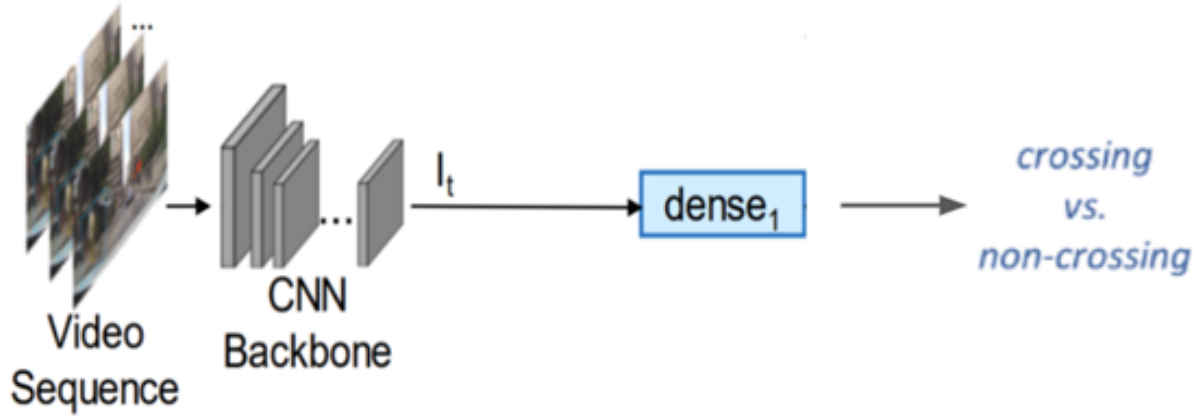

Figure 2: CNN-only model architecture

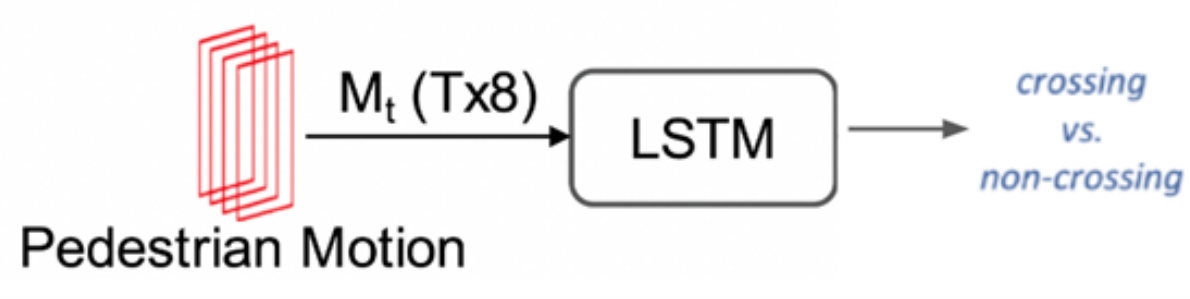

Figure 3: Motion encoder only architecture

Finally, we merged the pretrained models to construct our final hybrid model, as depicted in Figure 4.

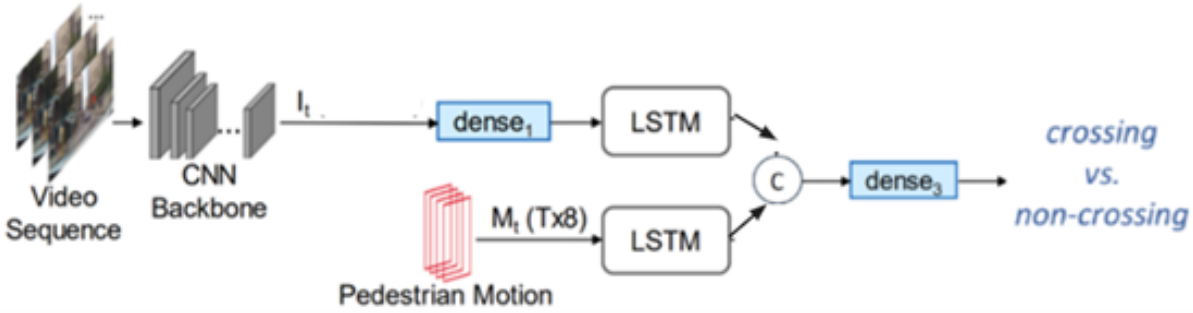

Figure 4: Hybrid model architecture

The results are shown in Table 2. In our experiments, the simple RNN Encoder performs better, meaning that the system is not able to combine the two modalities together. A possible direction towards improving the performance of the whole system

is to put in more effort into the optimization process of the CNN encoder and adapting it to the real-world street driving domain.

Table 2: Final results of three models

	test/f1
Hybrid model	0.8035
CNN Encoder	0.7808
RNN Encoder	0.812

Our overall best result (**86.57** test f1) was achieved with a hybrid system using 3 RNNs to encode each of the modalities (image, bounding box, behaviour).

References

[1] Smail Ait Bouhsain, Saeed Saadatnejad, and Alexandre Alahi. *Pedestrian Intention Prediction: A Multi-task Perspective*. 2021. arXiv: `2010.10270 [cs.CV]`.

[2] Amir Rasouli et al. "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 (ICCV)*. Oct. 2019.