Pedestrian Intent Prediction

Stefano Zanatta, Pasquale Coscia, Lamberto Ballan

January 7, 2021

1 Introduction

We implemented the work of Haziq Razali, Alexandre Alah: Pedestrian-Intention-Prediction[**PedestrianIP**]. The original work consists in a Convolutional Neural Network - LSTM, with the goal to predict the intention of pedestrians on crossing the road.
The model was originally trained on Lausanne Dataset (not public) and on JAAD Dataset[1]. The former was recorded by static cameras, the latter on moving cameras (placed in the cars).
The goal of this project is to study the pipeline, reproduce the experiments, adapt and improve the model for the JAAD and PIE datasets[2].

2 Lausanne Dataset

3 JAAD Dataset

The JAAD datasets consists of mp4 videos, each one associated to a rich description, frame to frame, of what is happening on a given moment:

- id of the pedestrian
- coordinates of the pedestrian on the relative frame
- standing, walking
- looking at the traffic
- hand gesture
- speeding up/slowing down
- crossing the road
- occlusion

[1] `http://data.nvision2.eecs.yorku.ca/JAAD_dataset/`
[2] `https://data.nvision2.eecs.yorku.ca/PIE_dataset/`

For the target we only use the information about crossing the road.

4 Pipeline

Adapting the ground truth

The first step is to create the ground truth for the model by transforming the input into a csv. Originally, the Lausanne dataset required a step to locate the pedestrians (with an external object detector) and the (static) coordinates of the road. This is not longer required, because JAAD and PIE include the coordinates of the pedestrians and whether or not they are crossing the road.

frame	id	tlx	tly	width	height	walking	standing	looking	incrossing
1,0	463	730	69	118	0	1	0	0	0
1,0	463	730	69	118	0	1	0	0	1

Trajectories

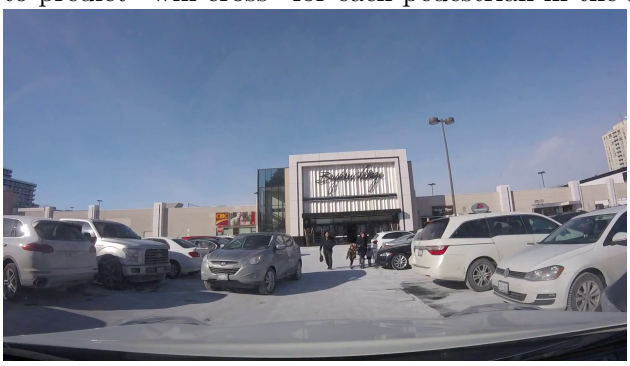
The second Step consists in assigning, at each frame, the "lifetime" of the pedestrian in the video, and a "cross" column, indicating if they ever crossed the road in their lifetime.

incrossing	lifetime	cross
0	568	0
0	568	0

Hungarian Tracker

Remove pedestrians with a lifetime below a threshold, and change pedestrians IDs to a numeric value. We had to change the algorithm that defined whether the pedestrian crossed or not the road, before it was designed for videos where the pedestrian always started from a (non-crossing)

side of the screen and then, eventually, he could cross the road or not. Now the new algorithm does not make any assumption on the initial position, to adapt to the JAAD and PIE datasets, where pedestrians can alternate the crossing of the road with the non-crossing and start in a crossing position. For example, this is the first frame of the Jaad video n5. In this case we expect the model to predict "will cross" for each pedestrian in the screen.



Crop Pedestrians

Crops each pedestrian into images, organized in sub-folders:

```
dataset
└── all
    └── crops
        └── name_of_the_video
            └── id_of_the_pedestrian
                └── id_pedestrian_png
```

Then appends the path of the crop to each frame's annotation.

filename	folderpath
00000001.png	crops_0001000000000
00000001.png	crops_0001000000000

Save Scenes

Add the whole scenes to the dataset and adds their paths to the annotations.

```
dataset
└── all
    └── scenes
        └── name_of_the_video
            └── id_video_png
```

The new annotations columns are:

scene_filename	scene_folderpath
00000001.png	scenes_0001
00000001.png	scenes_0001

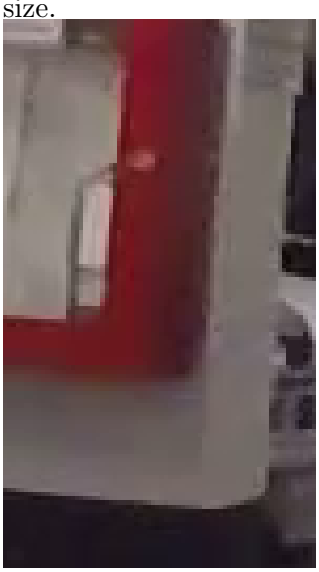
Split train test

In the last step the dataset is split into train and test:

```
dataset
└── train
dataset
└── test
```

5 Model

The model is composed of a VGG-16 pre trained CNN, a fully connected layer (feature embedder), a LSTM and a linear classifier to predict if the pedestrian crossed the road. Unless differently specified, we used 50 epochs, sequences of 8 observed frames for each prediction (obs_len), 32 batch size.



VGG-16

VGG-16 is a (pre-trained CNN) feature extractor. This operation is executed for each image in the batch.

Feature Embedder

Embed the CNN output into a 255 units vector. Then performs dropout (0.5).

LSTM

Process the sequence through the LSTM with ReLU activation function and dropout.

Linear Classifier

Classify whether the pedestrian crossed the road or not.

Changes to the original model

To improve the results, we added the entire scene as input for the prediction (JAADScene). We added a new VGG-16 CNN, followed by a LSTM to create the sequence, then we merged the scene-LSTM output with the crops-LSTM outputs and adapted the input size of the linear predictor. (#todo)

For another version of the model, we added more actions as target for the linear predictor. This makes the task harder, but should make the model more robust to overfitting.(#todo)

6 PIE Dataset

#todo

7 PIE Pipeline

#todo

8 Results

The model works best when there is a clear distinction between the pedestrians are not-crossing (for their initial lifetime) and crossing the road (at the end of their lifetime). It works poorly for large roads where the same pedestrians can move in and out the road repeatedly.
The original experiments where done on datasets where the direction of movement of the pedestrians where already known. For the Jaad dataset the directions were unknown and the results were much worse.

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

To get better results we had to remove the draw-trajectories min-lifetime thresholds (so a pedestrian is considered more often, even if they are crossing the road at frame 0 and leaves the screen shortly after). This change improved the results by increasing the dataset size.
The following results show the base model trained on Jaad dataset and the model improved with

the scenes input.
(all results are based on test data).

Model	Precision	Recall	Accuracy
Jaad	0.750	0.558	0.6714285714285714
Jaad+Scenes	0.742	0.535	0.7428571428571429

The results improved by adding the scenes, but the train accuracy (0.977) is still much higher, showing some overfitting.

Changing the pre-trained CNNs

We tested the same model changing the pretrained VGG-16 CNN (just with crops).

Model	Precision	Recall	Accuracy
VGG-16 CNN	0.750	0.558	0.6714285714285714
Wide Resnet50 2	0.622	0.535	0.6571428571428571
googlenet	0.733	0.512	0.6857142857142857
densenet201	0.565	0.302	0.6428571428571429

9 Other works on JAAD

We explored some works tested on the JAAD dataset by different authors.

Spatiotemporal Relationship Reasoning for Pedestrian Intent Prediction

They[**liu2020spatiotemporal**] based their network on Graph Convolutional Neural Networks[**GCNN**], by viewing the scene as a graph and then applying the same convolutions to different zones. The prediction accuracy was at its best at 30 frames (1 second), dropping by ˜10% after the 90^{th} and before the 10^{th} frame. In our work we predicted the next move after 8 frames, so we increased that to 30 for a comparison.

model	frames	Accuracy on K^{th} frame
Graph-CNN	30	76.98%
Graph-CNN	60	73.09%
Graph-CNN	60	68.31%
Crops+googlenet(ours)	30	%

PEDESTRIAN INTENTION PREDICTION: A MULTI-TASK PERSPECTIVE

This model is more similar to ours, because it inputs the pedestrian's cropped images to a feature extractor (CNN), a LSTM and to a fully connected layer for prediction. The main difference is that they perform multi-tasking learning, predicting both the crossing/non-crossing and the velocity of each bounding box (for each pedestrian). They also tested the same model adding the scene to the input (other then the crops), but reducing their size for faster training.
Adding a different task (velocity prediction) improved the intent prediction by ˜2%.

model	Accuracy
Single-task PV-LSTM	89.67
Multi-task PV-LSTM	91.48

mainbib