Pose-Based Pedestrian Crossing Intention Prediction

with Recurrent Graph Convolutions and Explainability-Driven Data Augmentation

Abel García Romera

Master Thesis Dissertation
Master's Degree in Computer Vision
Autonomous University of Barcelona

github.com/abel-gr/PedestrianCrossingIntention

July 2023

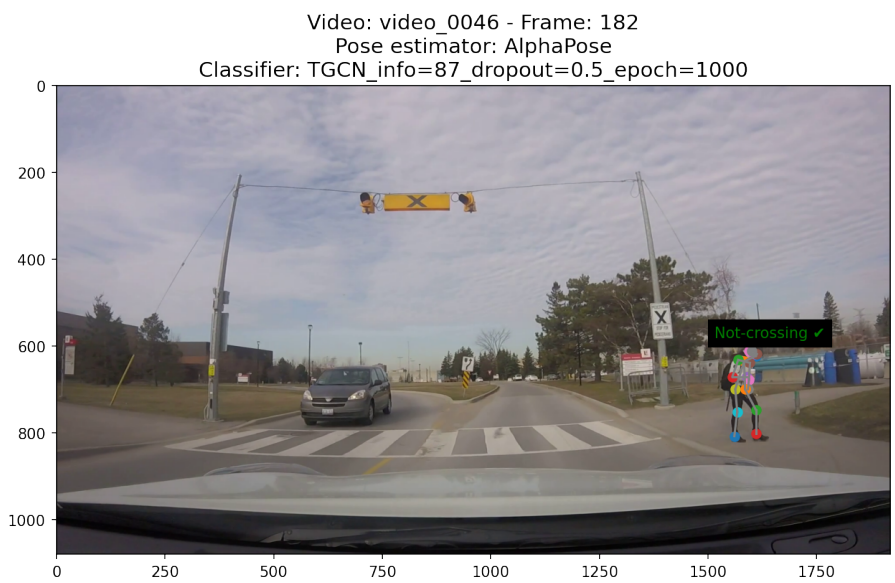
Table of Contents

Table of Contents

Introduction

Pedestrian Crossing Intention

Binary classification between the crossing and not-crossing labels for each pedestrian.



Introduction

Contributions:

- Spatial-Temporal Graph Neural Network that classifies the crossing intention of pedestrians based on their 2D skeleton coordinates.
- Explainability analysis using our own Grad-CAM [1] implementation for the Spatial-Temporal Graph Neural Network.
- Explainability-driven data augmentation technique that generates new skeletons.

Table of Contents

State of the art

State of the art review:

- Pedestrian crossing intention prediction papers.
- Explainability and data augmentation methods for crossing intention prediction.

State of the art

- Previous works rely on visual information from cameras [2].
- Incorporating pedestrian velocities and positions enhances accuracy [3, 4].
- Pose-based methods yield better results by using skeleton joints [5, 6, 7].
- Lightweight methods outperform complex approaches [8, 9].
- TrouSPI-Net uses spatio-temporal features and atrous convolutions [8].
- PCPA incorporates RNN branches and 3D convolutions [9].
- PedFormer proposes a cross-modal Transformer architecture [10].

Conclusion

Better results are achieved with methods that include human pose and temporal features.

State of the art

- Explainability for deep learning in Pedestrian Crossing Intention is still uncommon [11].
- Yao et al. propose an Attentive Relation Network (ARN) model that provides interpretable results based on detected traffic objects [11].
- Chen et al. generate text-based explanations of driver reasoning, but it is computationally expensive [12].
- Data augmentation in crossing intention prediction is very uncommon [13, 14].

Conclusion

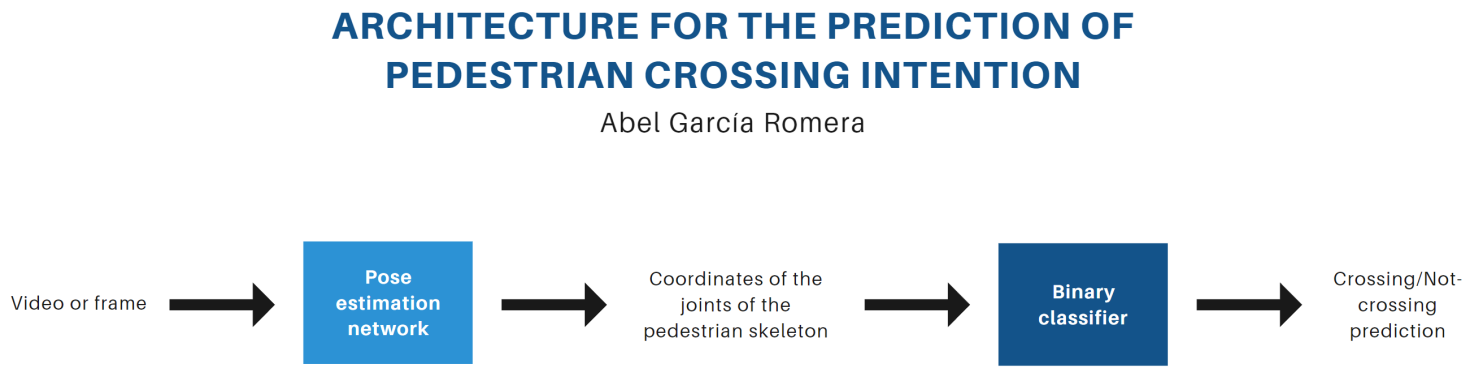
No found papers explore explainability or data augmentation using estimated pose for crossing intention prediction.

Table of Contents

Proposed architecture

The skeletons of the pedestrians are extracted from videos using a Pose Estimation network.

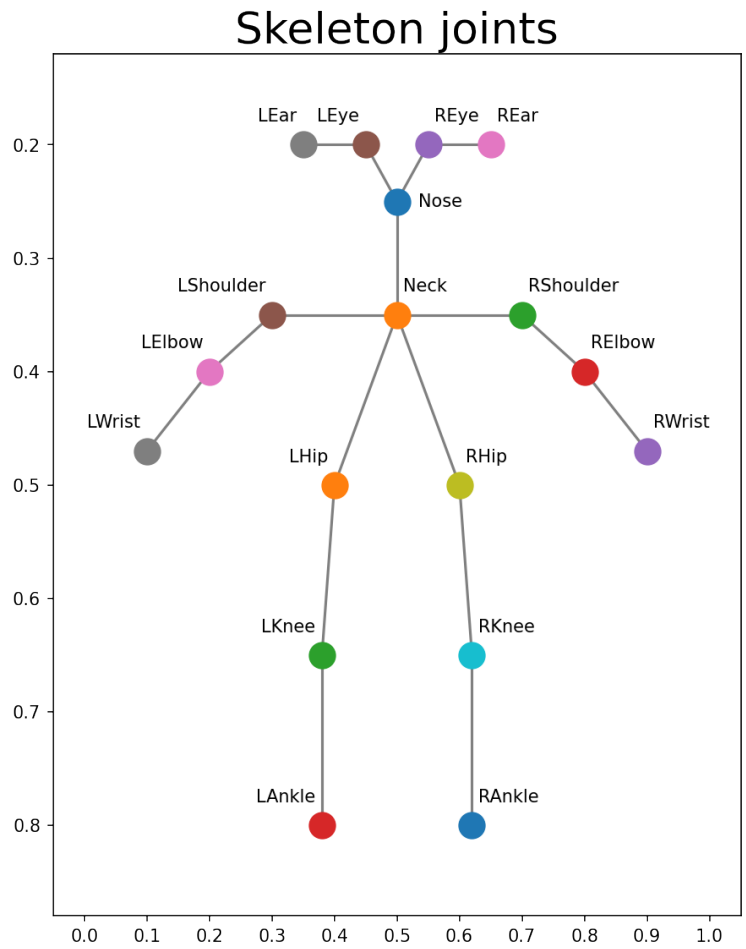
The coordinates of the joints of that skeleton form the input of a binary classifier that classifies between Crossing and Not-crossing.



Proposed architecture: Pose estimation network

Pose Estimation

Detection of the position and orientation of a person. In general this is done by predicting the coordinates of specific key points that are called joints [15].



Proposed architecture: Pose estimation network

There are chiefly two types of approaches to solve this task:

- Top-down approaches.
- Bottom-up approaches.

Proposed architecture: Pose estimation network

Top-down approach

Employ a person detector that allows to locate the people in each frame, and then for each of these people a single-person pose estimator is executed.

Top-down approach

Its execution time is proportional to the number of people that appear in the image. Another problem is that if the person detector fails, the Pose Estimation system will not be able to extract any skeleton of the person [15].

Proposed architecture: Pose estimation network

Bottom-up approach

This type of approach does not rely on any person detection, so it no longer suffers from either of the two problems of Top-down approaches.

Bottom-up approach

These methods tend to be slower to execute, since their final stages usually require longer inference times [16].

Proposed architecture: Pose estimation network

A new bottom-up Pose Estimation method appeared, capable of extracting the skeletons of multiple people with a time performance comparable to top-down approaches: OpenPose [17].

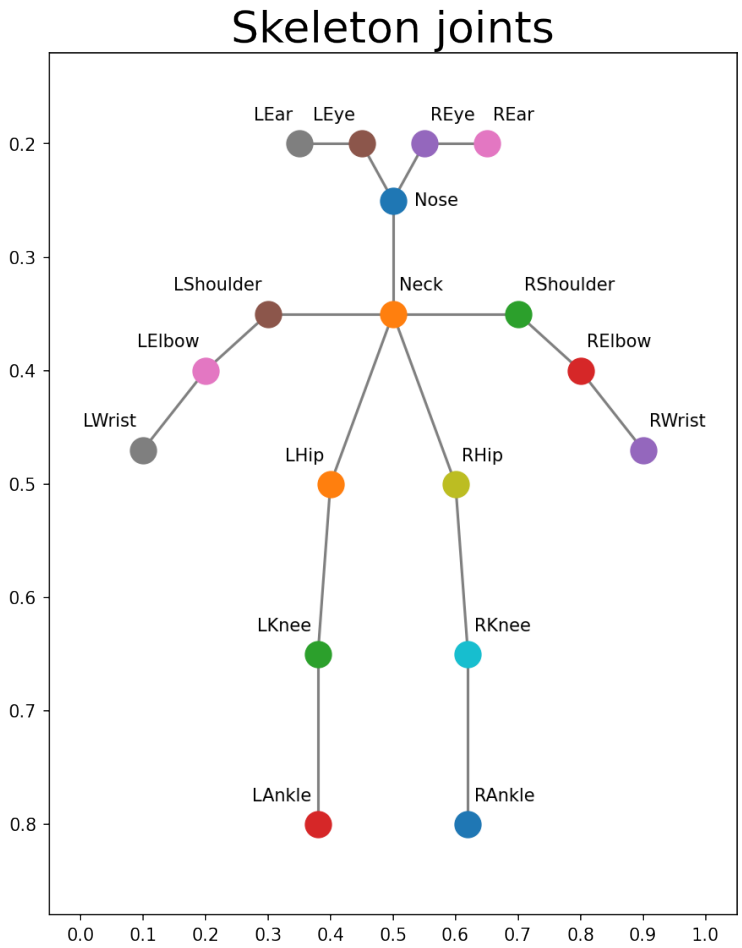
Nevertheless, despite the drawbacks of the top-down approaches, they provide fast results when the number of people present in the image is not large.

One of each will be used: as a top-down algorithm AlphaPose [18] will be utilized, and as a bottom-up method the selected option is OpenPose.

Proposed architecture: Binary classifier

We can model the body skeletons as a graph and process them using Graph Convolutional operators.

We can also include temporal information using Spatial-Temporal Graph Neural Networks (ST-GNNs) [19], whose input are a sequence of graphs corresponding to different time steps.



Proposed architecture: Binary classifier

Mainly there are two strategies to capture the temporal dependencies of multiple graphs:

- Adding a temporal convolutional layer using Convolutional Neural Networks.
- Using Recurrent Neural Networks (RNN). Specifically, the Recurrent Graph Convolutional Layers (RGCL) implement the convolutional operators of graphs but in the form of a RNN.

Proposed architecture: Binary classifier

The Recurrent Graph Convolutional Layers are the type of GNNs that are used in this work.

Specifically, the layers used are:

- Chebyshev Graph Convolutional Gated Recurrent Unit Cell (GConvGRU) [20].
- Chebyshev Graph Convolutional Long Short Term Memory Cell (GConvLSTM) [20].
- Temporal Graph Convolutional Gated Recurrent Cell (TGCN) [21].
- Graph Convolutional Long Short Term Memory Cell (GCLSTM) [22].

Proposed architecture: Binary classifier

Architecture proposal for the classifier.

Input shape (B_s, N_g, N_n, N_f).

- B_s is the batch size.
- N_g is the number of graphs (amount of previous frames to use).
- N_n is the number of nodes.
- N_f is the number of node features.

Layer	Input shape	Output shape
Recurrent part	$[B_s, N_g, 18, 3]$	-
RGCL	$[B_s, 18, 3]$	$[B_s, 18, 3]$
Dropout	$[B_s, 18, 3]$	$[B_s, 18, 3]$
ReLU	$[B_s, 18, 3]$	$[B_s, 18, 3]$
Non-recurrent part	$[B_s, 18, 3]$	-
Reshape	$[B_s, 18, 3]$	$[B_s, 54]$
Linear	$[B_s, 54]$	$[B_s, 37]$
Dropout	$[B_s, 37]$	$[B_s, 37]$
ReLU	$[B_s, 37]$	$[B_s, 37]$
Linear	$[B_s, 37]$	$[B_s, 18]$
Dropout	$[B_s, 18]$	$[B_s, 18]$
ReLU	$[B_s, 18]$	$[B_s, 18]$
Linear	$[B_s, 18]$	$[B_s, 2]$
Softmax	$[B_s, 2]$	$[B_s, 2]$

Table of Contents

Dataset

The dataset used in this work is the Joint Attention in Autonomous Driving dataset (JAAD).

It is a widely utilized dataset in the field of autonomous driving, particularly in research papers focusing on pedestrian behavior [15].

Dataset

The JAAD dataset contains 346 short videos between five and ten seconds in length that were extracted from almost 240 hours of driving footage recorded in North America and Eastern Europe [15].

Dataset

Each video contains annotations with the behavior and actions of pedestrians such as crossing, walking, standing, looking at the car, etc.

Total number of frames	82,032
Total number of annotated frames	82,032
Total number of pedestrians	2,786
Number of pedestrian bounding boxes	378,643
Number of pedestrians with behavior annotations	686
Average length of pedestrian track in frames	121

Dataset

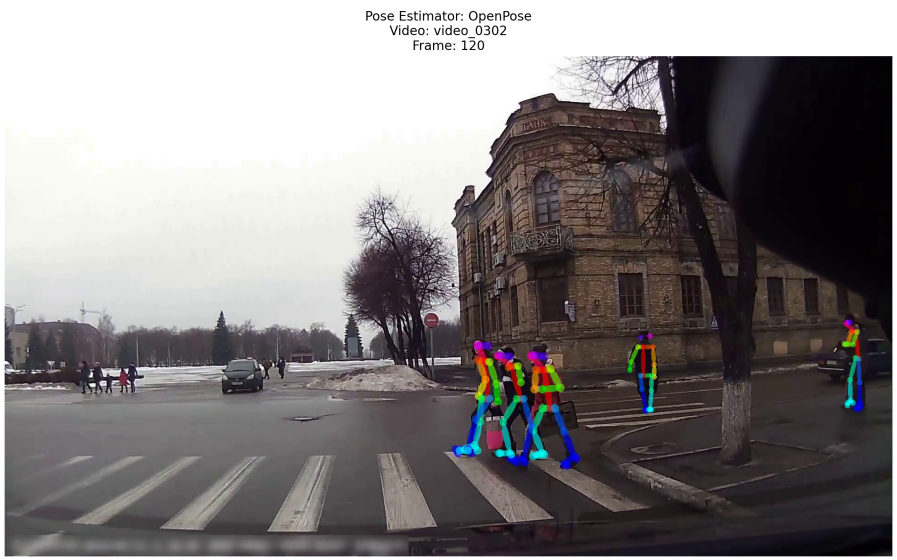
It can be seen the imbalance between classes in the three subsets.

	Train	Test	Val
Crossing	35,889	29,723	5,258
Not-Crossing	25,916	23,243	4,325
Total	61,805	52,966	9,583

Table of Contents

Pose Estimation

Pose Estimation results:



(a) Using OpenPose

(b) Using AlphaPose

Pose Estimation

Pose Estimation results:

(a) Using OpenPose

(b) Using AlphaPose

Pose Estimation

Pose Estimation results:

(a) Using OpenPose

(b) Using AlphaPose

Pose Estimation

Mean execution time per video to estimate the pose of the pedestrians averaged over all JAAD videos, both executed in standard Google Colab:

Pose estimator	Mean execution time
OpenPose	16.6 seconds
AlphaPose	45.4 seconds

Table of Contents

Quantitative analysis of pose estimation

Comparison of the total number of pedestrians with crossing/not-crossing annotations, and the number of detected pedestrians (estimated skeletons) by OpenPose and AlphaPose among all frames of all videos of JAAD dataset.

	OpenPose	AlphaPose
Total number of detected pedestrians	116,358 (93.57%)	122,801 (98.75%)
Total number of undetected pedestrians	7,996 (6.43%)	1,553 (1.25%)
Total number of pedestrians annotations	124,354 (100.00%)	124,354 (100.00%)

Quantitative analysis of pose estimation

For each range of percentages, number of skeletons estimated by each pose estimator in which as many joints percentage as indicated by the interval, were not detected.

Missing joints (%)	OpenPose	AlphaPose
0	27,314 (21.96%)	122,507 (98.51%)
(0, 10]	46,361 (37.29%)	294 (0.24%)
(10, 20]	25,017 (20.10%)	0 (0.00%)
(20, 30]	5,405 (4.35%)	0 (0.00%)
(30, 40]	5,852 (4.71%)	0 (0.00%)
(40, 50]	3,115 (2.50%)	0 (0.00%)
(50, 60]	1,862 (1.50%)	0 (0.00%)
(60, 70]	577 (0.46%)	0 (0.00%)
(70, 80]	352 (0.28%)	0 (0.00%)
(80, 90]	481 (0.39%)	0 (0.00%)
(90, 100)	21 (0.02%)	0 (0.00%)
100	7,996 (6.43%)	1,553 (1.25%)
Total	124,354 (100.00%)	124,354 (100.00%)

Quantitative analysis of pose estimation

For each range of percentages, number of skeletons estimated by each pose estimator in which as many joints percentage as indicated by the interval, were not detected.

Missing joints (%)	OpenPose	AlphaPose
0	27,314 (21.96%)	122,507 (98.51%)
(0, 50]	85,750 (68.96%)	294 (0.24%)
(50, 100)	3,293 (2.65%)	0 (0.00%)
100	7,996 (6.43%)	1,553 (1.25%)
Total	124,354 (100.00%)	124,354 (100.00%)

Quantitative analysis of pose estimation

Number of skeletons estimated by each pose estimator in which as many joints as indicated, were not estimated.

Undetected skeletons that would have all joints missing are not included.

Number of missing joints	OpenPose	AlphaPose
0	27,314 (23.47%)	122,507 (99.76%)
1	29,631 (25.47%)	294 (0.24%)
2	16,730 (14.38%)	0 (0%)
3	15,062 (12.94%)	0 (0%)
4	5,618 (4.83%)	0 (0%)
5	4,337 (3.73%)	0 (0%)
More than 5	17,666 (15.18%)	0 (0%)
Detected pedestrians	116,358 (100.00%)	122,801 (100.00%)

Crossing intention classification

Hyperparameters of the classifier that are used in the Grid Search to find the optimal configuration:

Hyperparameter	Values used in Grid Search
Input skeletons (Pose estimator used)	OpenPose, AlphaPose
Temporal info. (Number of input frames)	$\{n \mid n \in [3, 120],\ n \equiv 0 \pmod 3\}$
Graph layer	GConvGRU, GConvLSTM, TGCN, GCLSTM
Dropout	0.3, 0.5, 0.7

Crossing intention classification

For each set of skeletons estimated by each pose estimator, the hyperparameters that have given the best **validation accuracy** using these skeletons as input to the classifier are shown. Both classifiers have been trained for 100 epochs.

Pose estimator	Optimal classifier hyperparameters			Classifier training and validation metrics			
				Train subset		Validation subset	
	Graph Layer	Temporal Info	Dropout	Accuracy	f1-score	Accuracy	f1-score
OpenPose	GCLSTM	81	0.3	0.7714	0.8376	0.7668	0.8248
AlphaPose	TGCN	87	0.5	0.7588	0.8403	0.8071	0.8609

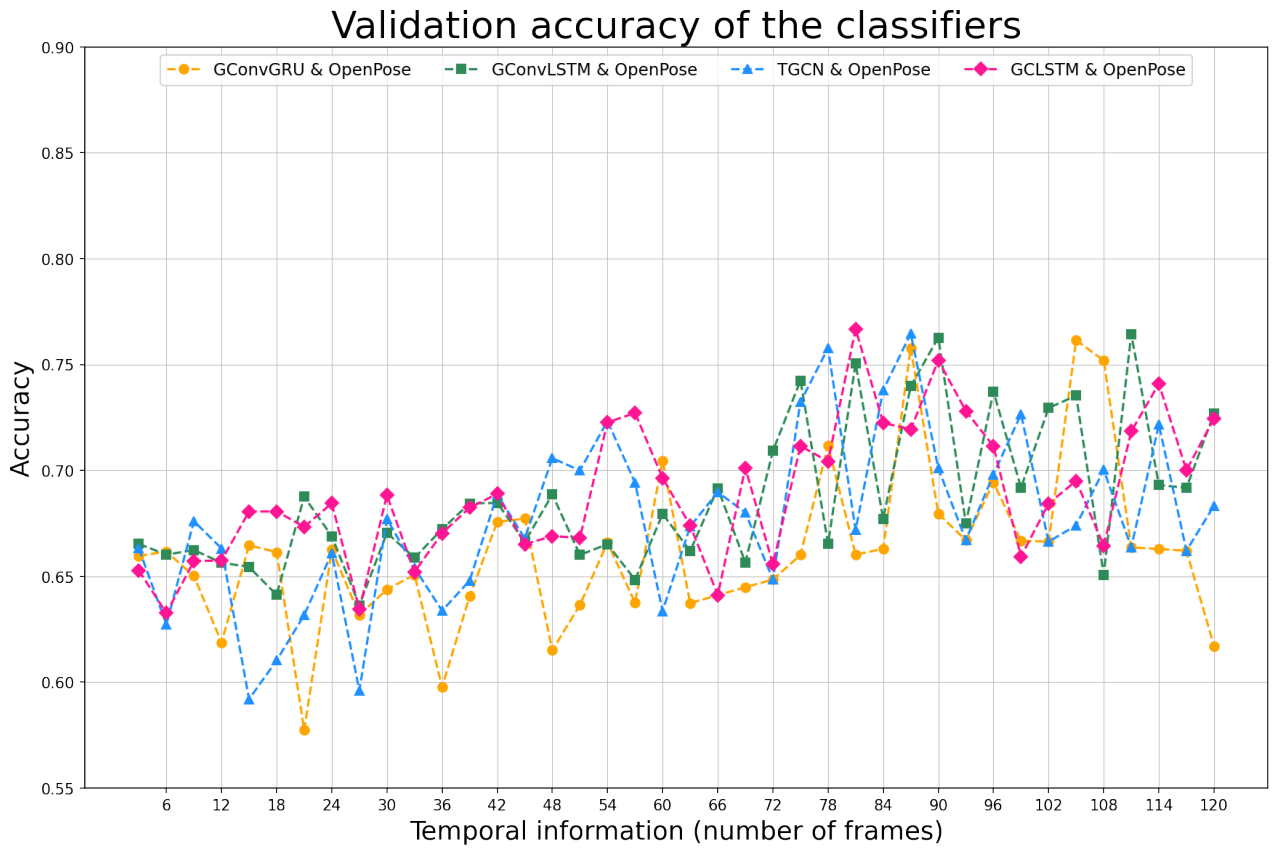
Crossing intention classification

For each set of skeletons estimated by each pose estimator, the hyperparameters that have given the best **validation f1-score** using these skeletons as input to the classifier are shown. Both classifiers have been trained for 100 epochs.

Pose estimator	Optimal classifier hyperparameters			Classifier training and validation metrics			
				Train subset		Validation subset	
	Graph Layer	Temporal Info	Dropout	Accuracy	f1-score	Accuracy	f1-score
OpenPose	GConvGRU	87	0.3	0.7235	0.8275	0.7579	0.8373
AlphaPose	TGCN	87	0.5	0.7588	0.8403	0.8071	0.8609

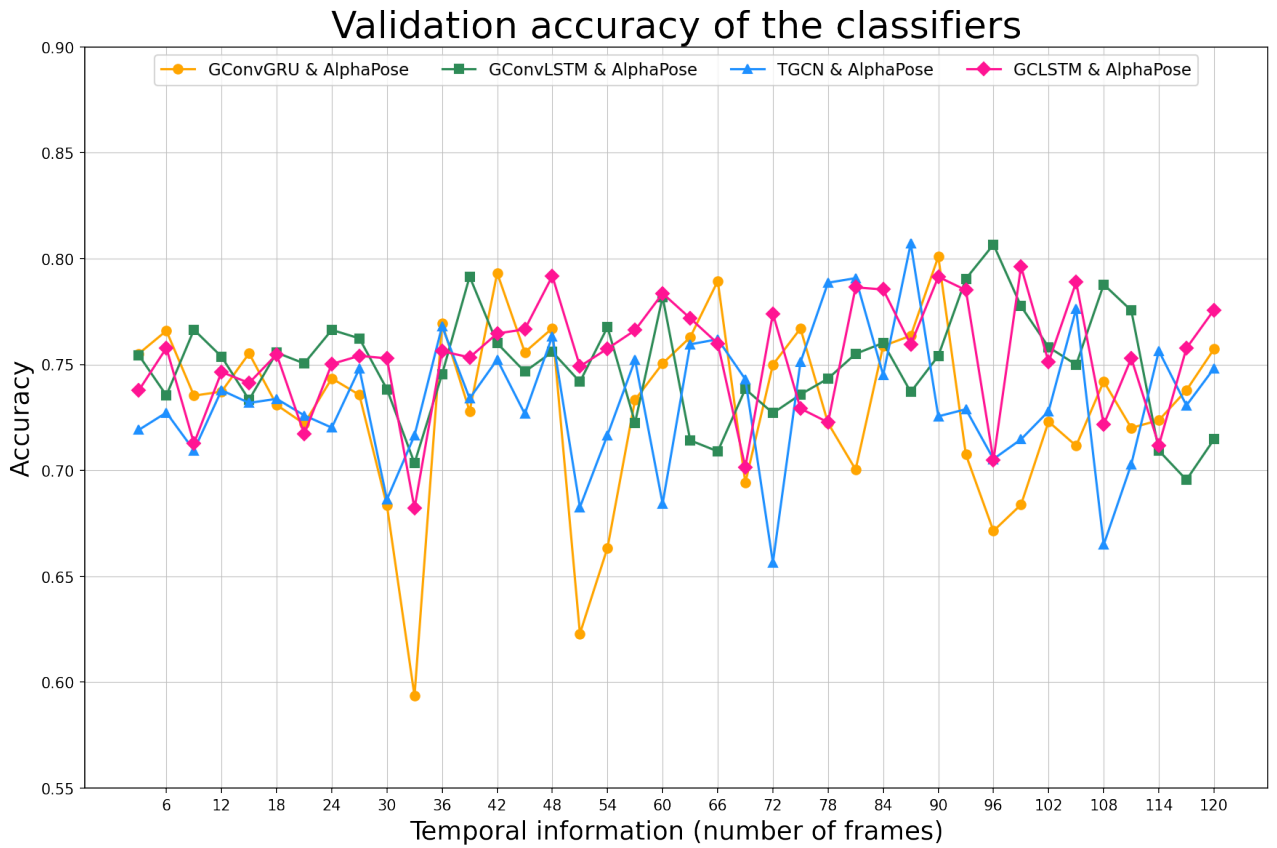
Crossing intention classification

Validation accuracy based on the number of frames used by each classifier in its input, using OpenPose skeletons as input:



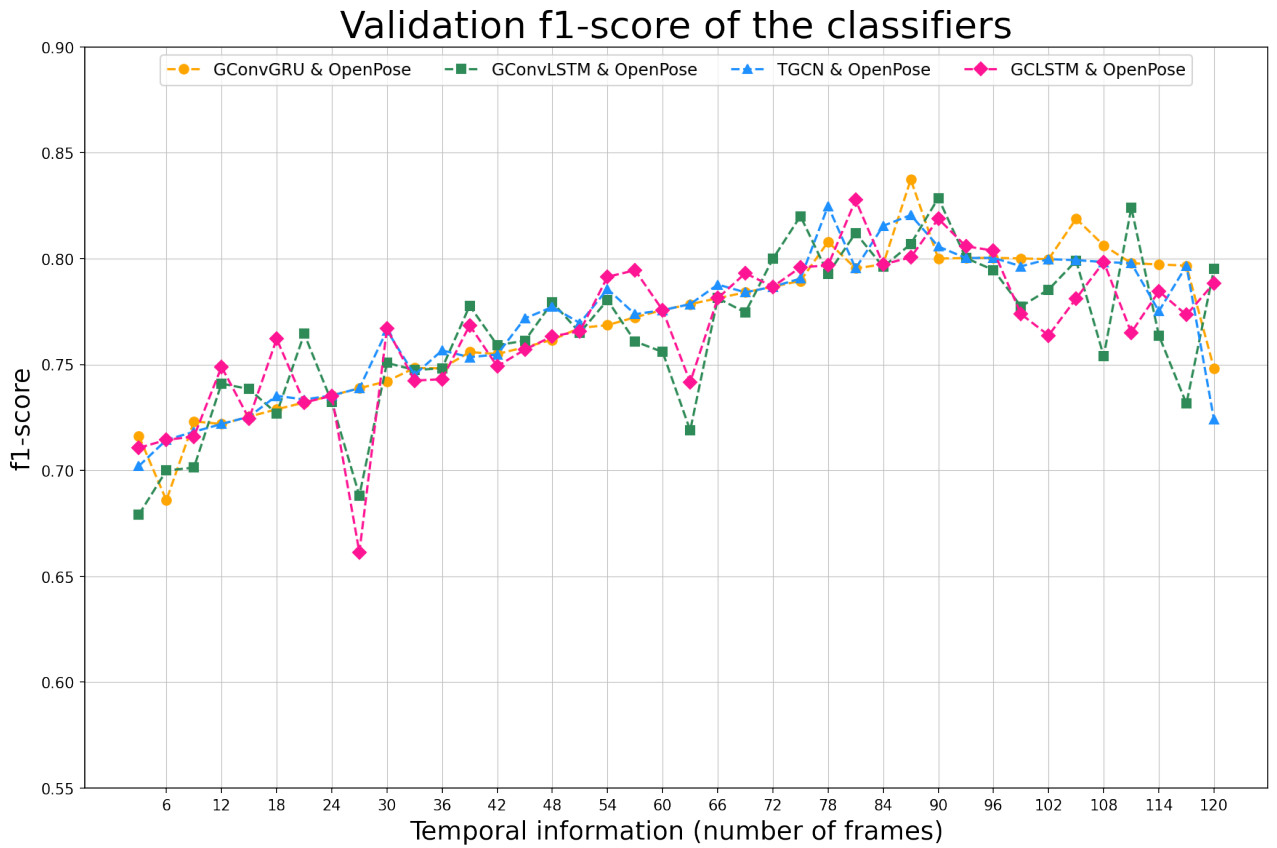
Crossing intention classification

Validation accuracy based on the number of frames used by each classifier in its input, using AlphaPose skeletons as input:



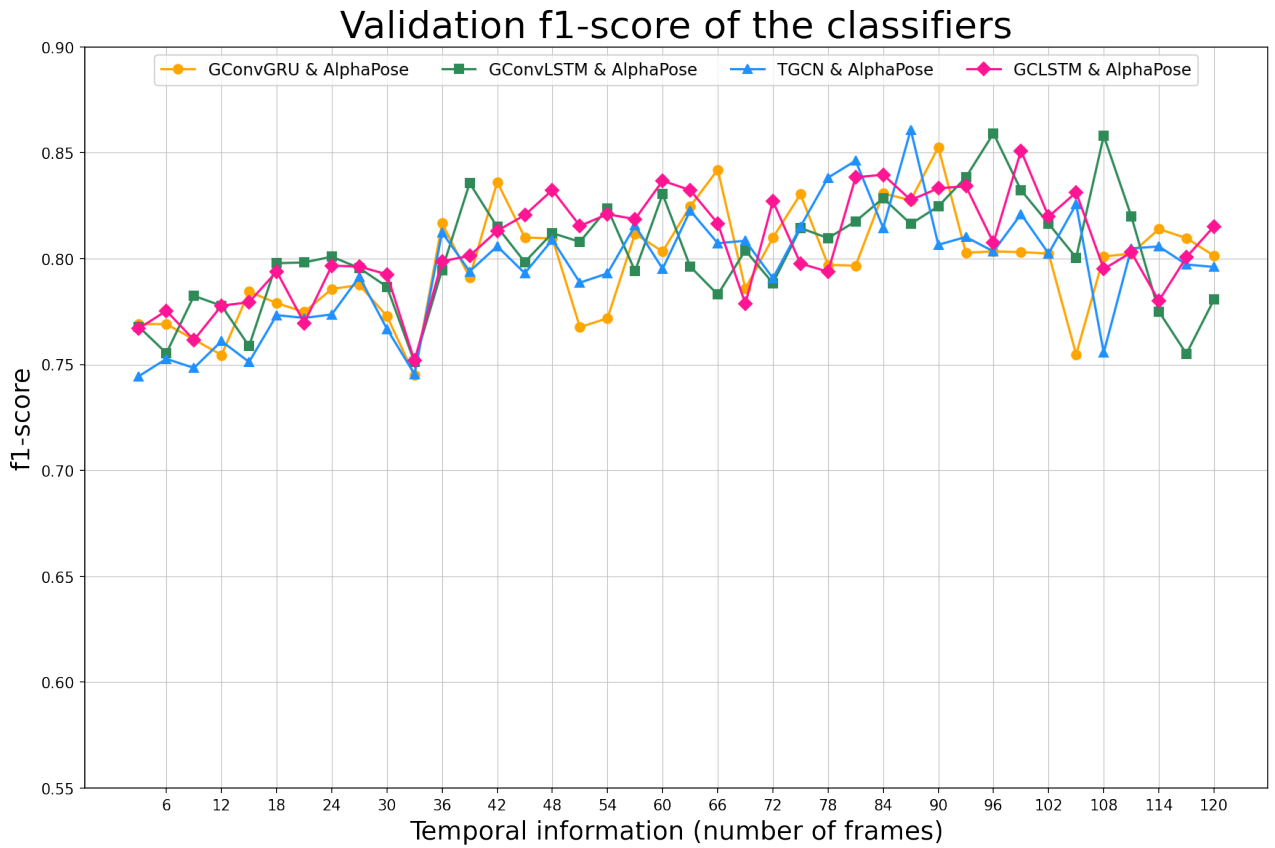
Crossing intention classification

Validation f1-score based on the number of frames used by each classifier in its input, using OpenPose skeletons as input:



Crossing intention classification

Validation f1-score based on the number of frames used by each classifier in its input, using AlphaPose skeletons as input:



Explainability

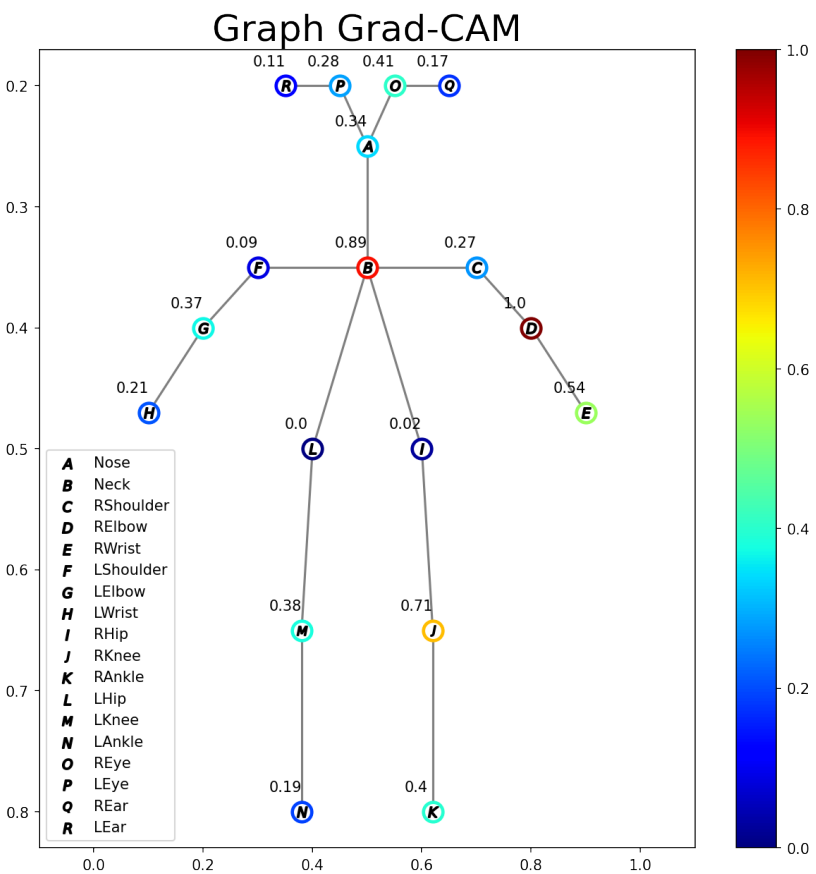
- Explainability analysis aimed at understanding how the classifier generates the predictions of the crossing intention.
- One way to use explainability methods in the context of GNNs is to identify the most important nodes for predicting the pedestrian's intention.

Explainability

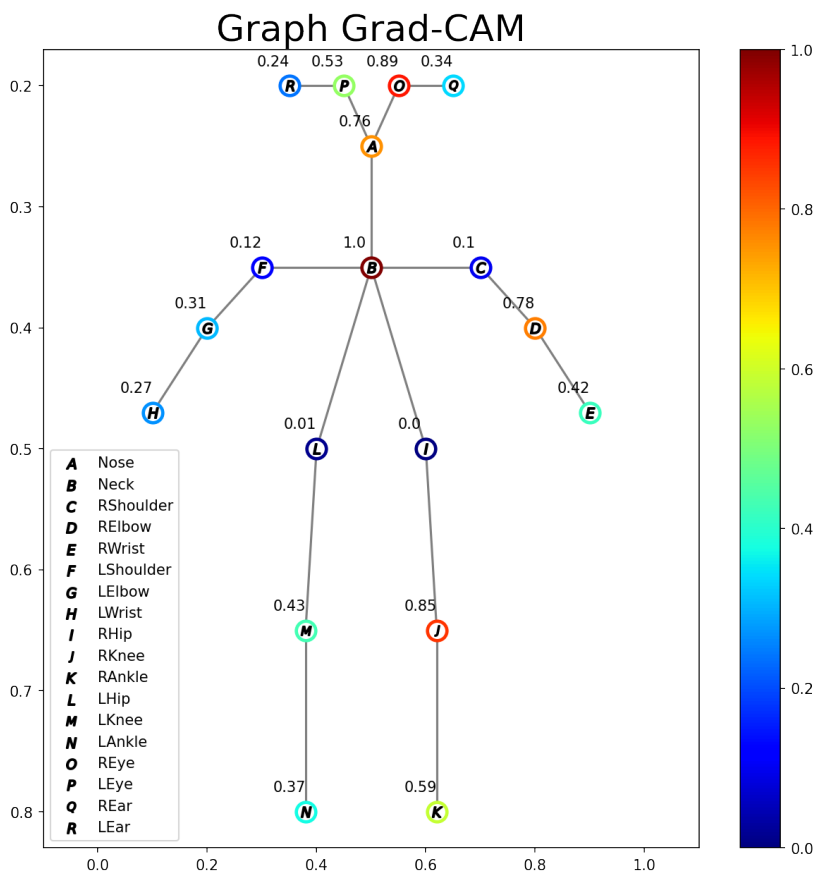
- Grad-CAM (Gradient-weighted Class Activation Mapping) [1] is a technique that generates a heatmap that highlights the regions of an input image that are important for the model to produce a prediction.
- Grad-CAM does not require retraining the network.
- Instead, Grad-CAM computes the gradient of the output of the network with respect to the feature maps of a convolutional layer.
- In the case of our proposed architecture the classifier input are not images, but pedestrian skeletons instead.
- It has been implemented a new version of Grad-CAM that utilizes the activations of Recurrent Graph Convolutional Layers (RGCL) instead of traditional convolutional layers.
- This results in a heatmap of the same size as the number of output nodes of the RGCL.

Explainability

Heatmap resulting of Grad-Cam execution **over all skeletons** in the dataset. Higher values mean more important joints:



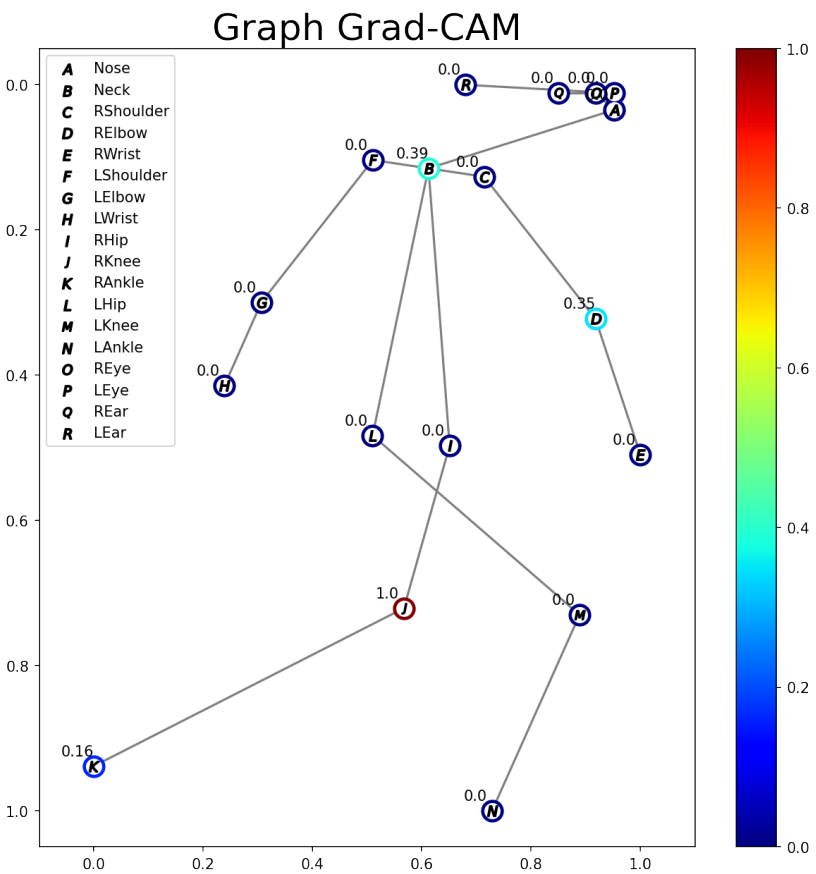
(a) Not-crossing class



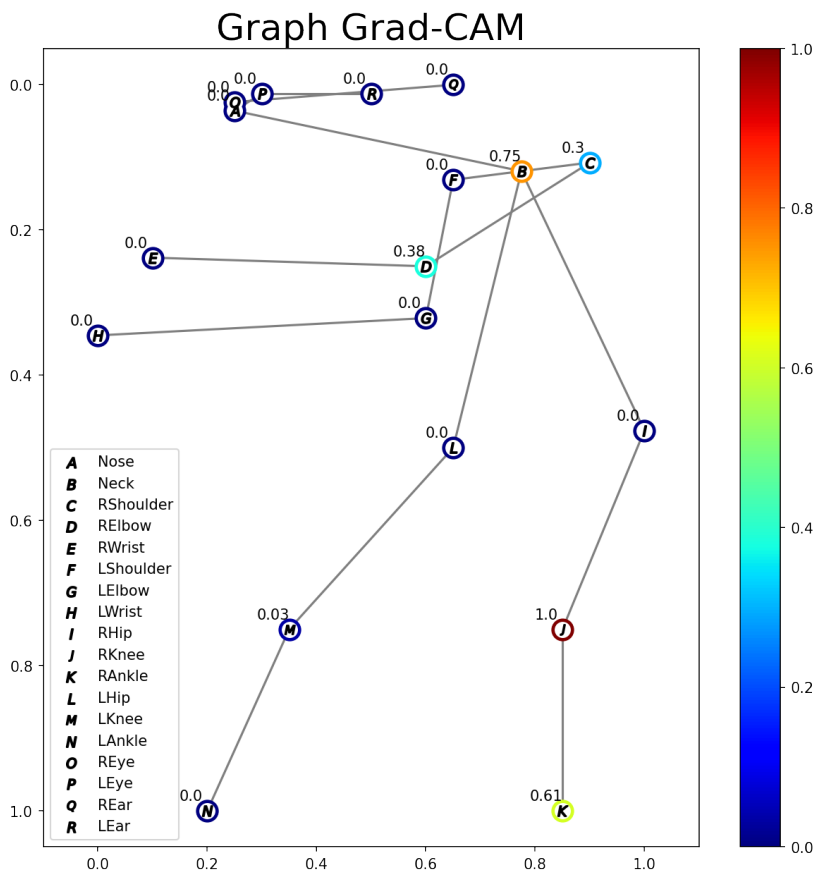
(b) Crossing class

Explainability

Heatmap resulting of Grad-Cam execution **for two specific skeletons** of the dataset. Higher values mean more important joints:



(a) Crossing from left to right



(b) Crossing from right to left

Data augmentation

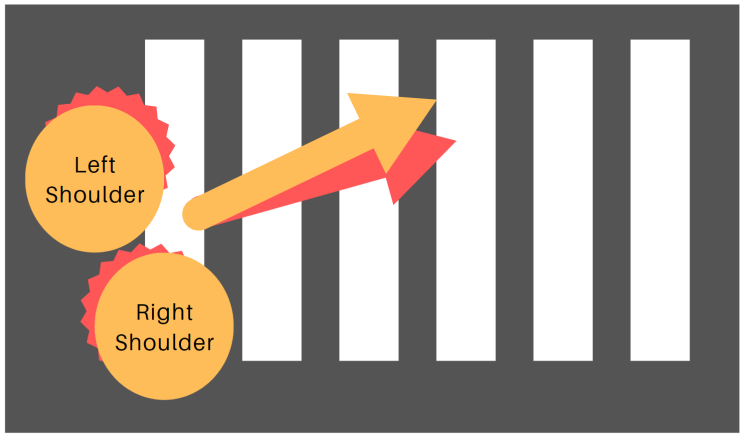
- The proposed data augmentation technique operates on the estimated skeletons of pedestrians.
- The insights gained from the explainability analysis are utilized to design an explainability-driven data augmentation technique that generates new skeletons by modifying the training subset.
- The objective is to generate more meaningful and varied skeletons.

Data augmentation

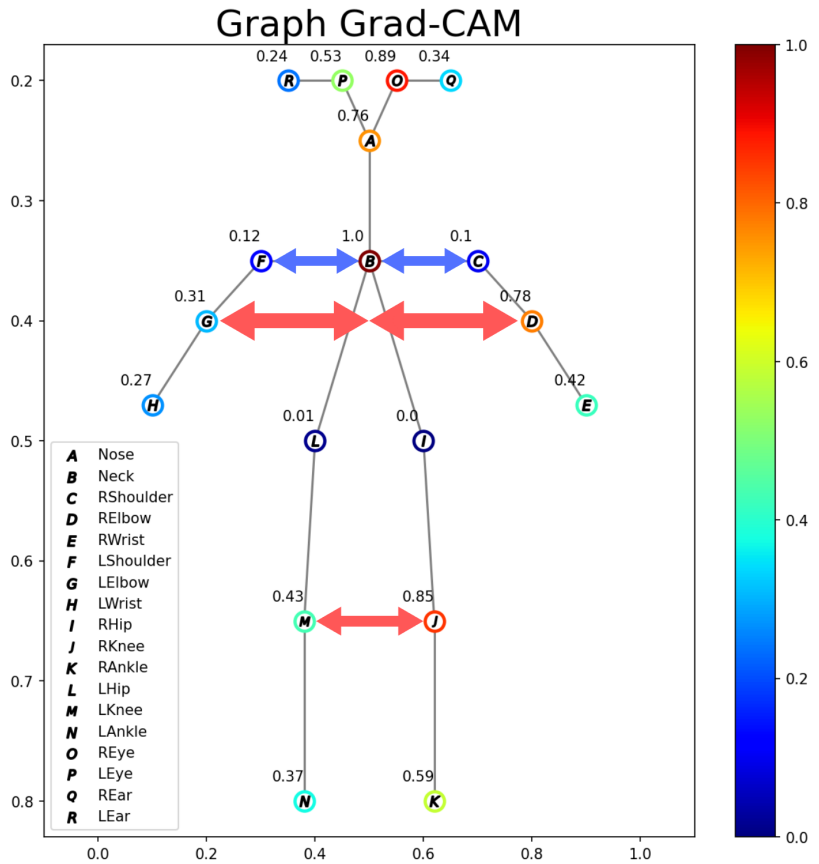
3D rotation is simulated by increasing or decreasing the distance on the horizontal axis with respect to the center of the body on that axis:

Top view of the street

(Not used in practice, only for theoretical explanation)



(a) Top view



(b) Front view

Data augmentation

Metrics of the classifier using different types of data augmentation:

Data augmentation	Perturbed joints	Perturbation range	Classifier training and validation metrics			
			Train subset		Validation subset	
			Accuracy	f1-score	Accuracy	f1-score
No augmentation	-	-	0.7588	0.8403	0.8071	0.8609
Random multiplicative noise	All joints	[0.95, 1.05]	0.7750	0.8453	0.6326	0.7077
Random multiplicative noise	Neck	[0.95, 1.05]	0.6927	0.8184	0.6675	0.8006
Random multiplicative noise	All joints	[0.98, 1.02]	0.7380	0.8366	0.7101	0.8163
Random multiplicative noise	Neck	[0.98, 1.02]	0.7910	0.8606	0.7112	0.8031
Horizontal displacement	LShoulder, RShoulder	[0.98, 1.02]	0.7292	0.8318	0.7032	0.8119
Horizontal displacement	LElbow, RElbow	[0.98, 1.02]	0.7911	0.8550	**0.8226**	**0.8659**
Horizontal displacement	LKnee, RKnee	[0.98, 1.02]	**0.7973**	**0.8635**	0.7223	0.8073
Horizontal displacement	LElbow, RElbow, LKnee, RKnee	[0.98, 1.02]	0.7812	0.8505	0.7769	0.8343

Table of Contents

Final results

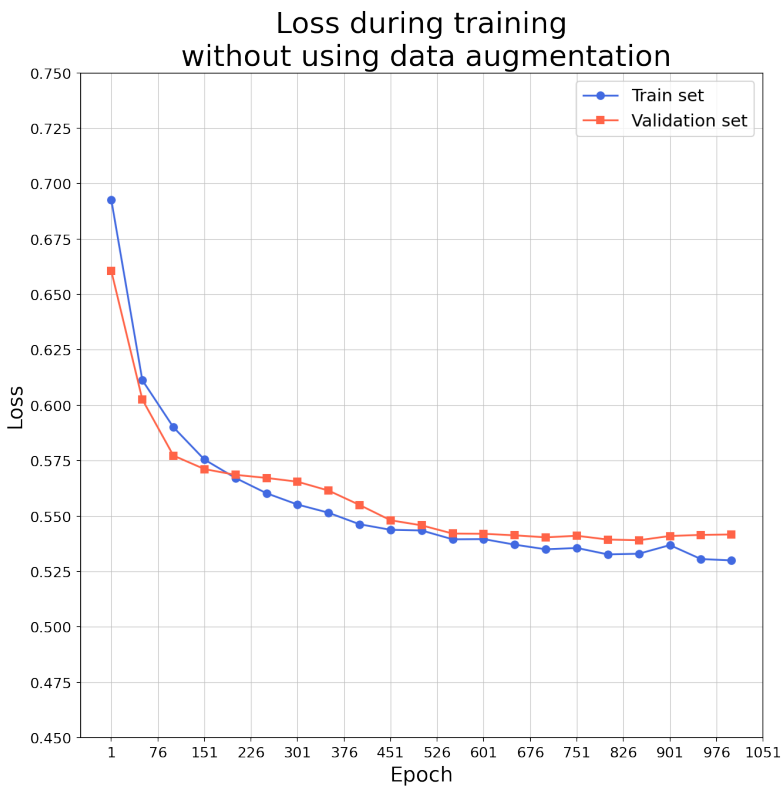
Hyperparameters that have given the best validation metrics in the Grid Search:

Pose estimator (classifier input skeletons)	Optimal classifier hyperparameters		
	Graph Layer	Temporal Info	Dropout
AlphaPose	TGCN	87	0.5

Final results

That model has been retrained but using 1000 epochs instead of 100 as in the Grid Search.

Loss plots during training for 1000 epochs:



(a) Without data augmentation

(b) LElbow and RElbow displacement

Final results: Quantitative results

Comparison to the state-of-the-art methods for the test subset classification metrics of the pedestrian crossing intention prediction:

Method	Accuracy	ROC AUC	F1-score	Precision	Recall
ATGC	0.64	0.60	0.53	0.50	0.56
MM-LSTM	0.80	0.60	0.40	0.39	0.41
I3D	0.82	0.75	0.55	0.49	0.63
SF-GRU	0.83	0.77	0.58	0.51	0.67
PCPA	0.83	0.77	0.57	0.50	0.66
BiPed	0.83	**0.79**	0.60	0.52	0.71
PedFormer	**0.93**	0.76	0.54	0.65	0.46
PedRGCN without D. Augm. (ours)	0.76	0.77	0.83	0.78	0.89
PedRGCN with explainability-driven D. Augm. (ours)	0.76	0.78	**0.84**	**0.79**	**0.90**

Final results: Quantitative results

- Six trainings and tests have been performed both with and without data augmentation.
- We have computed the classification metrics for each execution and utilized them in a Wilcoxon signed-rank test.
- The resulting p-value has been 0.03125 for all the metrics, which is lower than the significance level of 0.05.
- Consequently, it can be concluded that the proposed explainability-driven data augmentation indeed improves the metrics.

Final results: Qualitative results

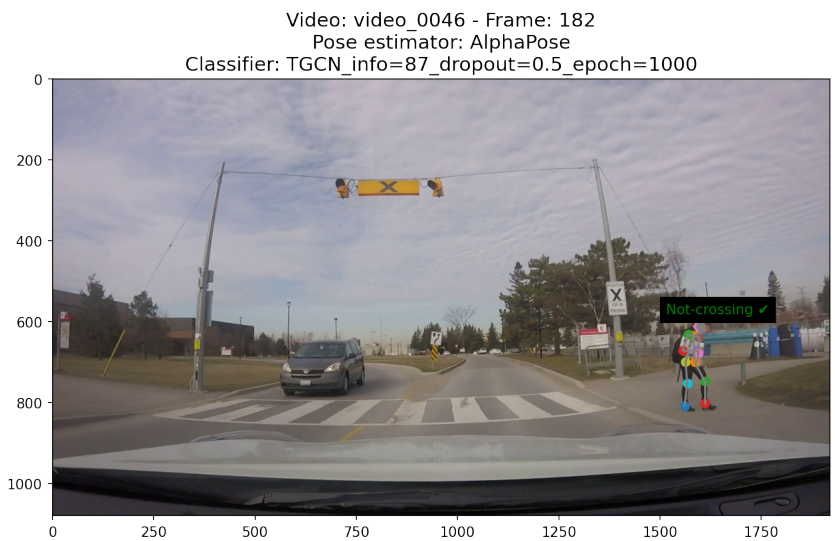
Classifier prediction for frame 168 of video 55:

(a) Trained without data augmentation

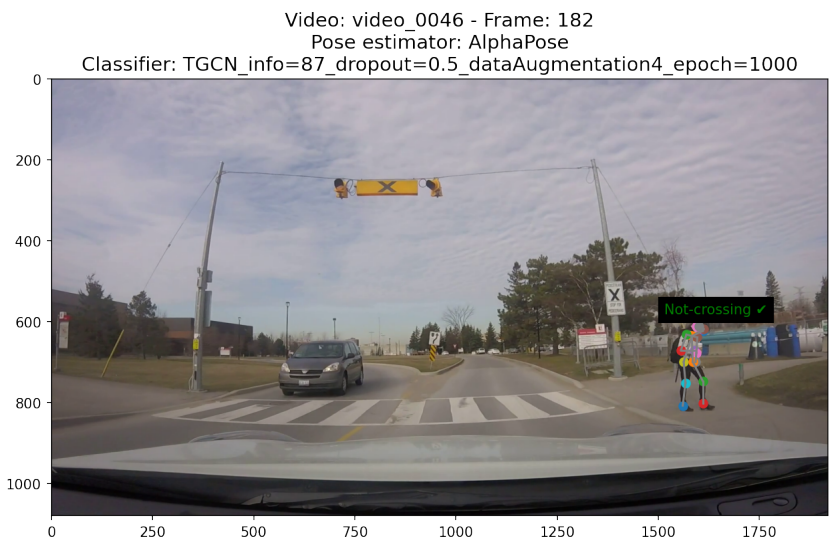
(b) Trained with data augmentation

Final results: Qualitative results

Classifier prediction for frame 182 of video 46:



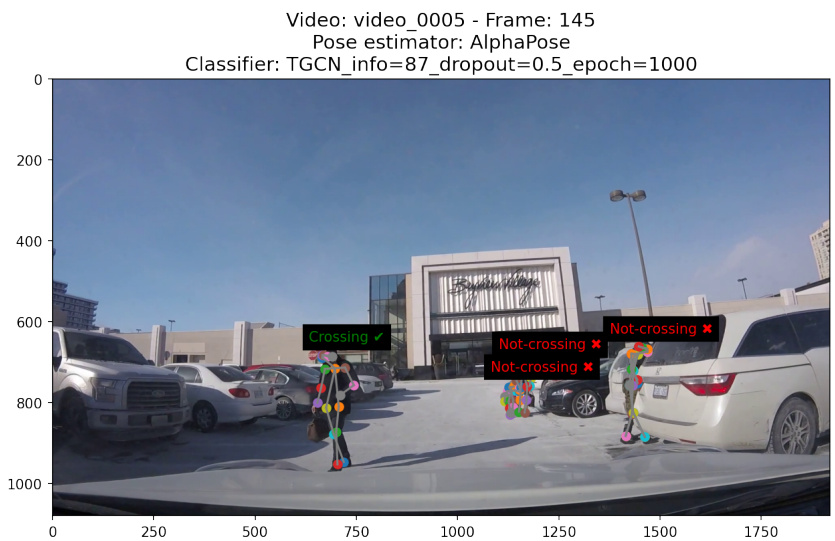
(a) Trained without data augmentation



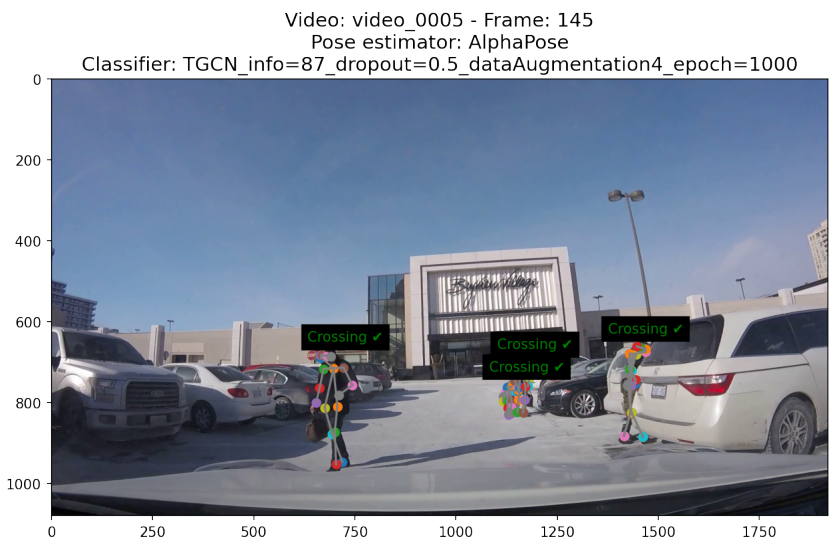
(b) Trained with data augmentation

Final results: Qualitative results

Classifier prediction for frame 145 of video 5:



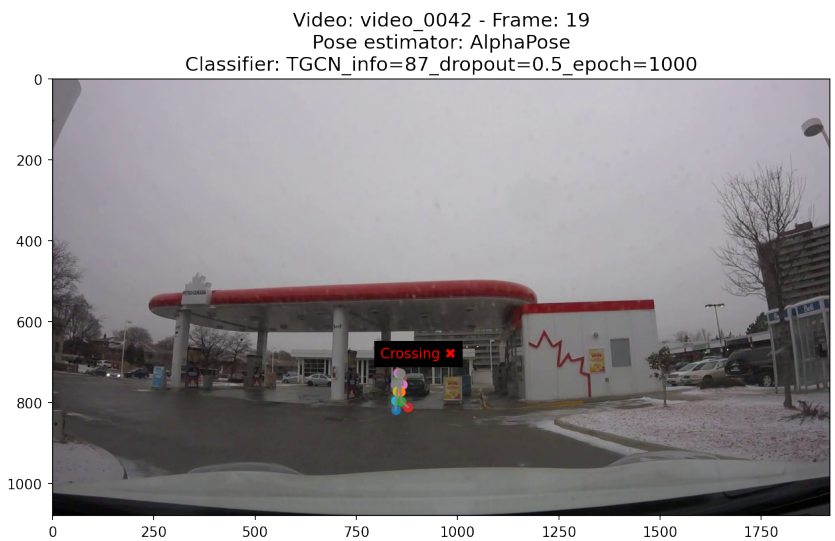
(a) Trained without data augmentation



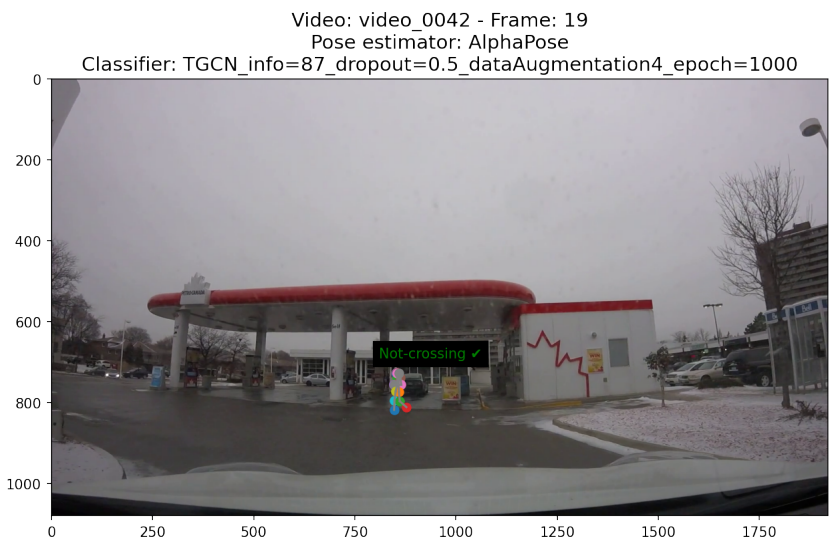
(b) Trained with data augmentation

Final results: Qualitative results

Classifier prediction for frame 19 of video 42:



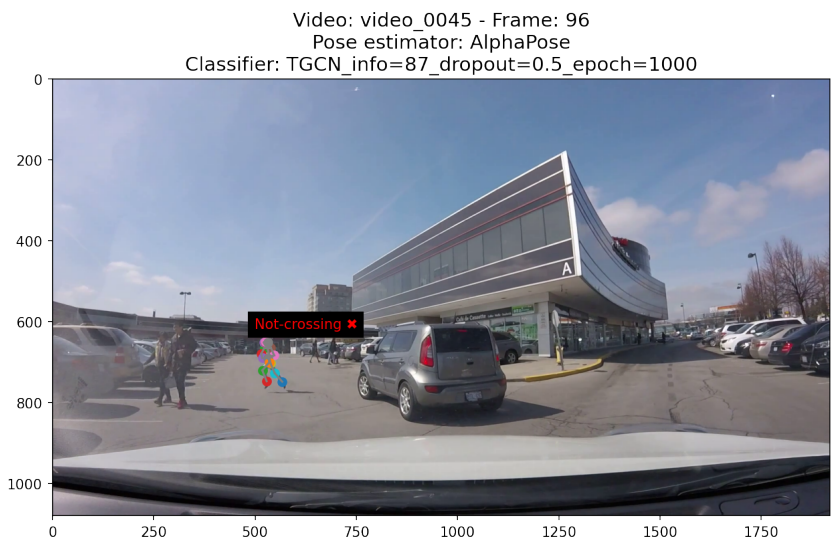
(a) Trained without data augmentation



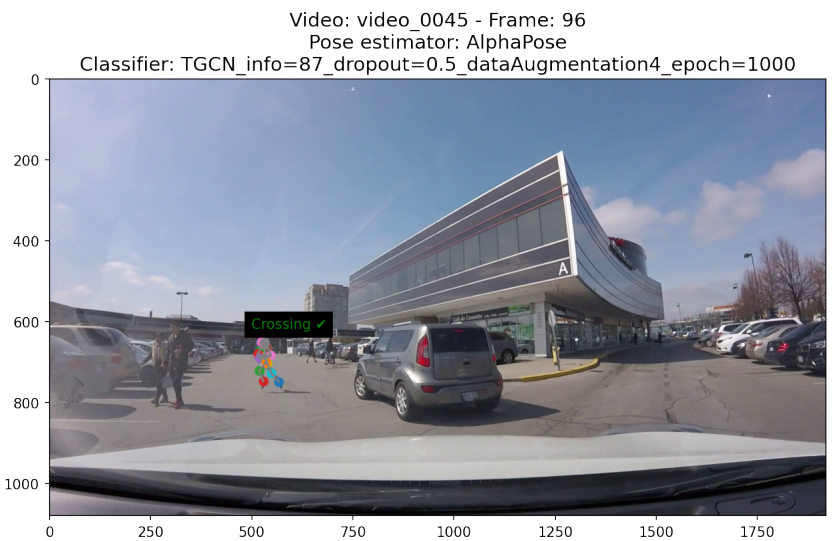
(b) Trained with data augmentation

Final results: Qualitative results

Classifier prediction for frame 96 of video 45:



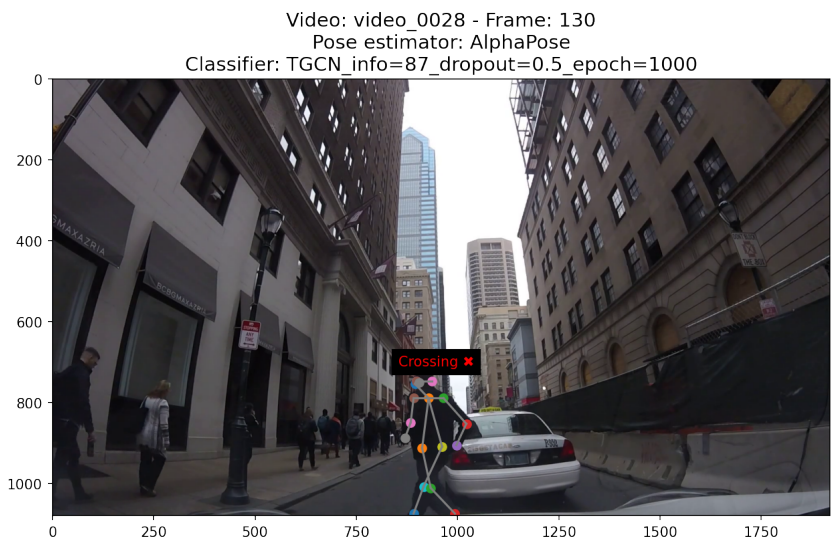
(a) Trained without data augmentation



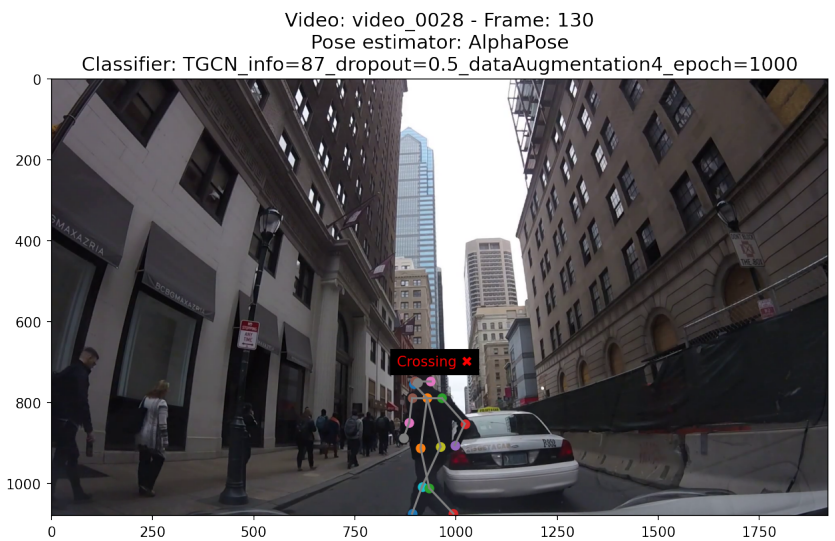
(b) Trained with data augmentation

Final results: Qualitative results

Classifier prediction for frame 130 of video 28:



(a) Trained without data augmentation



(b) Trained with data augmentation

Table of Contents

Conclusions

- We propose an architecture for predicting the crossing intention of pedestrians that outperforms the state-of-the-art methods in terms of f1-score, precision, and recall.
- The classifier's metrics using the skeletons estimated by AlphaPose as input were higher compared to those achieved using OpenPose.
- The classifier uses a Recurrent Graph Convolutional Layer (RGCL). The optimal architecture utilizes the TGCN layer and 87 input frames.
- An explainability analysis has been performed using our own implementation of Grad-Cam on the RGCL, which determined that the most critical joints are the Neck, Right Knee, and Right Elbow.
- Based on the results of the explainability analysis, we have proposed a data augmentation method that generates new skeletons for the training subset.
- It has been determined that applying random horizontal displacements to both the Right Knee and Left Knee enhance the metrics.

Conclusions

In the future, the results could be further improved by proposing new data augmentation methods based on the performed explainability analysis.

Additionally, the proposed pose-based model could be combined with visual models that provide a context of the scene from the car's point of view.

Table of Contents

References I

[1] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, oct 2019. [Online]. Available: https://arxiv.org/abs/1610.02391

[2] K. Saleh, M. Hossny, and S. Nahavandi, "Real-time intent prediction of pedestrians for autonomous ground vehicles via spatio-temporal densenet," 2019.

[3] S. A. Bouhsain, S. Saadatnejad, and A. Alahi, "Pedestrian intention prediction: A multi-task perspective," 2021.

[4] A. Rasouli, M. Rohani, and J. Luo, "Bifold and semantic reasoning for pedestrian behavior prediction," 2021. [Online]. Available: https://arxiv.org/abs/2012.03298

[5] Z. Fang and A. M. López, "Is the pedestrian going to cross? answering by 2d pose estimation," 2018.

[6] S. Zhang, M. Abdel-Aty, Y. Wu, and O. Zheng, "Pedestrian crossing intention prediction at red-light using pose estimation," April 2021. [Online]. Available: https://shilezhang.github.io/files/paper4.pdf

[7] H. Zhang, Y. Liu, C. Wang, R. Fu, Q. Sun, and Z. Li, "Research on a pedestrian crossing intention recognition model based on natura observation data," March 2020. [Online]. Available: https://www.researchgate.net/publication/340144653_Research_on_a_Pedestrian_Crossing_Intention_Recognition_Model_Based_on_Natural_Observation_Data

[8] J. Gesnouin, S. Pechberti, B. Stanciulescu, and F. Moutarde, "Trouspi-net: Spatio-temporal attention on parallel atrous convolutions and u-grus for skeletal pedestrian crossing prediction," 2021. [Online]. Available: https://arxiv.org/abs/2109.00953

[9] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Benchmark for Evaluating Pedestrian Action Prediction," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1258–1268. [Online]. Available: https://openaccess.thecvf.com/content/WACV2021/papers/Kotseruba_Benchmark_for_Evaluating_Pedestrian_Action_Prediction_WACV_2021_paper.pdf

[10] A. Rasouli and I. Kotseruba, "Pedformer: Pedestrian behavior prediction via cross-modal attention modulation and gated multitask learning," 2022. [Online]. Available: https://arxiv.org/abs/2210.07886

References II

[11] Y. Yao, E. Atkins, M. J. Roberson, R. Vasudevan, and X. Du, "Coupling intent and action for pedestrian crossing behavior prediction," 2021. [Online]. Available: https://arxiv.org/abs/2105.04133

[12] T. Chen, T. Jing, R. Tian, Y. Chen, J. Domeyer, H. Toyoda, R. Sherony, and Z. Ding, "Psi: A pedestrian behavior dataset for socially intelligent autonomous car," 2022. [Online]. Available: https://arxiv.org/abs/2112.02604

[13] J. Lorenzo, I. P. Alonso, R. Izquierdo, A. L. Ballardini, Álvaro Hernández Saz, D. F. Llorca, and M. Ángel Sotelo, "Capformer: Pedestrian crossing action prediction using transformer," 2021. [Online]. Available: https://doi.org/10.3390/s21175694

[14] S. Zamboni, Z. T. Kefato, S. Girdzijauskas, N. Christoffer, and L. D. Col, "Pedestrian trajectory prediction with convolutional neural networks," 2021. [Online]. Available: https://arxiv.org/abs/2010.05796

[15] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," October 2017. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2017_workshops/papers/w3/Rasouli_Are_They_Going_ICCV_2017_paper.pdf

[16] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," November 2016. [Online]. Available: https://arxiv.org/pdf/1605.03170.pdf

[17] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. [Online]. Available: https://arxiv.org/pdf/1812.08008v1.pdf

[18] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. [Online]. Available: https://arxiv.org/pdf/2211.03375.pdf

[19] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018. [Online]. Available: https://arxiv.org/abs/1801.07455

References III

[20] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," 2016. [Online]. Available: https://arxiv.org/pdf/1612.07659.pdf

[21] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, September 2020. [Online]. Available: https://arxiv.org/pdf/1811.05320.pdf

[22] J. Chen, X. Wang, and X. Xu, "Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction," October 2021. [Online]. Available: https://arxiv.org/pdf/1812.04206.pdf