HAL
open science

Multi-Task Cross-Modality Deep Learning for Pedestrian Risk Estimation

Danut Ovidiu Pop

To cite this version:

Danut Ovidiu Pop. Multi-Task Cross-Modality Deep Learning for Pedestrian Risk Estimation. Computer Science [cs]. Normandie Université; Babeş-Bolyai Universite, 2019. English. NNT : . tel-02493160v2

HAL Id: tel-02493160

https://theses.hal.science/tel-02493160v2

Submitted on 27 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PhD Thesis

Computer Science

Intelligent Transportation System

Multi-Task Cross-Modality Deep Learning for Pedestrian Risk Estimation

Defended by:

ENG. DĂNUŢ OVIDIU POP

Jury:

DR. Samia BOUCHAFA	Professor - Université de Evry, Paris Saclay, France	Examiner
DR. Fabien MOUTARDE	Professor- Ecole des Mines Paris, France	Examiner
DR. Fabrice MERIAUDEAU	Professor - Université de Bourgogne, France	Reviewer
DR. Mihaela Elena BREABAN	Associate Professor - Alexandru Ioan Cuza University, Romania	Reviewer
DR. Fawzi NASHASHIBI	Professor - Inria Paris, France	PhD Director
DR. Abdelaziz BENSRHAIR	Professor - INSA Rouen, France	PhD Director
DR. Horia F POP	Professor- Babeş-Bolyai University, Romania	PhD Director
DR. Alexandrina ROGOZAN	Associate Professor- INSA Rouen	PhD Advisor

8 November 2019
Rouen

Acknowledgements

First of all, I wish to address my sincere thanks to God, for His love, help, guidance, and protection during my Ph.D. period. Only with God's support, I was able to bring this success into my life.

Throughout my Ph.D. period, I have received a great deal of support and assistance from marvelous persons.

I would like to assign my thankfulness to my Ph.D. directors: Prof. Dr. Fawzi Nashashibi from Inria Paris, Rits Team, Paris, France and Prof. Dr. Abdelaziz Bensrhair from INSA Rouen, LITIS, STI Team, Rouen, France; for their guidance, patience, scientific and financial support during my Ph.D. studies. Without their help, I would not have been able to complete this work.

I would be glad to express my sincere gratitude to my Ph.D. director Prof. Dr. Horia F. Pop from Babeş-Bolyai University - Department of Computer Science, Cluj Napoca, Romania for the continuous support of my Ph.D. study, for his patience motivation and for linking me up with INSA.

I would be happy to give my special thanks to Assoc. Prof. Dr. Alexandrina Rogozan, who has also guided me through my research, encouraged me and who made possible the collaboration with INRIA, INSA and UBB. The door to Assoc. Prof. Dr. Alexandrina Rogozan office was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this thesis to be my work but steered me in the right direction whenever she thought I needed it.

My sincere thanks also go to Prof. Dr. Fawzi Nashashibi, and Dr. Prof. Dr. Abdelaziz Bensrhair, who provided me an opportunity to join their teams as a Ph.D. student, and who gave access to the laboratories and research facilities. Without their precious support, it would not be possible to conduct this research. It was great to feel part of such teams.

I would like to thank to the jury: Assoc. Prof. Dr. Mihaela Elena Breaban from Alexandru Ioan Cuza University, Iasi, Romania; Prof. Dr. Fabrice Meriaudeau from Université de Bourgogne, France; Prof. Dr. Fabien Moutarde from Ecole des Mines Paris, France and Prof. Dr. Samia Bouchafa from Université de Evry, Paris Saclay, France for the time and effort they spent on the review. Due to their comments, opinions and suggestions, our contribution has been much improved.

Besides my advisor, I would like to thank the rest of my thesis committee from Babeş-Bolyai University - Department of Computer Science, Cluj Napoca, Romania: Prof. Dr. Gabriela Czibula, Prof. Dr. Laura Diosan, and Prof. Dr. Lehel Csato, for their insightful comments and encouragement, but also for the hard questions which incented me to widen my research from various perspectives.

Being part of several academical institutes, I had the opportunity to meet amazing persons who helped me in a professional or personal way.

I am very grateful to Richard James from Inria Paris for his continuous guid-

ance in my manuscripts. I wish to acknowledge the help provided by Assoc. Prof. Dr.Clement Chatelain from INSA Rouen. His assistance was greatly appreciated.

I also want to thank my friends from Rouen: Kamal, Roger, Sahar, Linda, Mario, and Federica, for their help and friendship.

Last, but not least, I would be delighted to express my very profound gratitude to my lovely wife Cristina, she was always there for me, and to my families for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

Contents v

List of figures vii

List of Tables xi

1 Cross-Modality Pedestrian Recognition 9
1.1 Introduction 11
1.2 Related Work 14
1.2.1 Handcrafted Features Models 14
1.2.2 Deep Learning Features Models 15
1.2.3 Handcrafted Features Models vs. Deep Learning Features Models 16
1.3 Proposed Architectures for Pedestrian Classier 17
1.3.1 Classical Learning Approach 19
1.3.2 LeNet+ Convolutional Neural Network Architecture 20
1.3.3 Early Fusion architecture 21
1.3.4 Late Fusion architecture 21
1.3.5 Cross-Modality Learning Approaches 22
1.3.6 Late Fusion Pedestrian Classification with Incremental Cross-Modality Learning 25
1.4 Evaluation of Classification Components 26
1.4.1 Experimental Setups and Evaluation Protocol 26
1.4.2 Evaluation of Uni-Modal Learning Classifiers 27
1.4.3 Evaluation of the Particular Cross-Modality Learning Classifier . 28
1.4.4 Comparison of Uni-Modal Classifiers with Cross-Modality Learning Models 29
1.4.5 Early-Fusion vs Late-Fusion with Classical Learning method . . 33
1.4.6 Late-Fusion with Classical vs Cross-Modality Learning using LeNet and LeNet+ CNN architectures 34
1.4.7 Late-Fusion with Classical vs Incremental Cross-Modality Learning using AlexNet and VGG-16 CNN architectures 35
1.4.8 Comparisons with the state-of-the-art methods 36
1.5 Conclusion 39

2 Pedestrian Detection with Action Classification 41
2.1 Introduction 43
2.2 Related Work 46
2.2.1 Object Detectors 46
2.3 Pedestrian Detection 51
2.3.1 Pedestrian Detection Component 51

2.3.2 Depth Modality from JAAD Dataset 52
2.3.3 Optical Flow Modality from JAAD Dataset 53
2.4 Experiments 54
2.4.1 Data setup 54
2.4.2 Training protocol 55
2.4.3 The convolution neural network setups 56
2.4.4 Testing protocol 57
2.4.5 Evaluation protocol 57
2.5 Evaluation and Results 58
2.5.1 Evaluation of the Uni-Modal Pedestrian Detection Component . 58
2.5.2 Evaluation on Uni-Modal Incremental Cross-Modality Deep Learning Pedestrian Detection 59
2.5.3 Evaluation of Uni Modal Pedestrian Action Detection 61
2.5.4 Evaluation of Incremental Cross Modality Deep Learnig Pedestrian Action Detection 63
2.5.5 Comparison of the Uni-Modal vs Incremental Cross Modality Deep Learning Pedestrian Detection for Pedestrian Action Detection 64
2.6 Conclusion 70

3 Pedestrian Action Prediction and Time to Cross Estimation 73
3.1 Introduction 75
3.2 Related Work 78
3.2.1 Prediction Analysis Models 78
3.2.2 Related Studies Concerning Pedestrian Action Prediction 81
3.3 Method 83
3.3.1 Pedestrian Position and Action Prediction 83
3.3.2 Estimation of Time to Cross 84
3.4 Experiments 86
3.4.1 Data setup 86
3.4.2 Training protocol 86
3.4.3 Testing protocol 90
3.4.4 Evaluation protocol 91
3.5 Results 92
3.5.1 Evaluation of Pedestrian Actions Prediction 93
3.5.2 Evaluation of Pedestrian time to cross Component 93
3.6 Conclusion 101

4 Conclusion 103

A Annexes I
A.1 RTMAPS Architecture I

List of figures

1 The main architecture of our system. 5

1.1 The main architecture of our system. In red are the issues investigated in this Chapter. 12

1.2 The Handcrafted Features Model Architecture. The ROI Processing represents the regions of interest from the proposed image, which inserts it in a feature extraction algorithm and then inserts this emphasizes information in the learning classifier process. Whenever the learning process is over; it returns the trained classifier, which could be used in the real-time application. 16

1.3 The Deep Learning Features Model Architecture. The CNN extract implicit the features from the image which are learnt in the same step. When the learning stage is completed, it returns the trained classifier, which could be used in the real-time application. 17

1.4 The classical learning approach uses the same image modality for the training, validation, and testing processes. The Particular Cross-Modality learning uses the same image modality for training and validation, but a different one for testing. The Separate Cross-Modality learning uses the same image modality for training and testing, but a different one for validation. 19

1.5 The proposed extended LeNet Architecture (LeNet+). The extension consists in adding a ReLu and an LRN layer at the first Pooling layer, adding a Dropout layer at the first FC layer, using the Gaussian instead of Xavier algorithm for the weight filler and increasing the outputs for the first FC layer from 500 to 4096. 20

1.6 The Early Fusion Architecture 21

1.7 The MLP Late-Fusion Architecture 22

1.8 Correlated Cross-Modality Learning. The learning data consists of Multi-Modal Correlated images presented successively to the CNN for training and respectively in Multi-Modal or Uni-Modal images for validation. I∈{I_1,I_2 ... I_n}; D∈{D_1,D_2 ... D_n}; F∈{F_1,F_2 ... F_n}; I=Intensity; D=Depth; F=Optical Flow. 23

1.9 Incremental Cross-Modality Learning. The first CNN is learning (training + validation) on the same image modality. When the learning process is over, the weights information from the previous CNN is transferred to the next CNN in which the learning process starts with a different image modality. 24

1.10 The Late Fusion Architecture with the Incremental Cross-Modality Learning. The modality probabilistic output scores of Intensity (Pr(I)), Depth (Pr(D)) and Optical Flow (Pr(F)). 25

1.11 Single-modality vs. multi-modality ROC classification performance on Daimler testing dataset using images of 36 x 84 pixels. FPR at 90% detection rate. 33
1.12 The ROC classification performance on Daimler testing dataset; where L=LeNet Architecture, L+= Extended LeNet Architecture, CL=Classical Learning method, HO=Holdout validation, SM=Same Settings, SP=Specific Settings, K-Cross=K-fold Cross-Validation. 38

2.1 The main architecture of our system. In red are the issues investigated in the first chapter. In blue is the problems studied in this chapter. . . . 44
2.2 Pedestrian detection using the same tag for all pedestrians. P=Pedestrian 50
2.3 Pedestrian detection using multiple tags. PPC: Pedestrian is Preparing to Cross the street; PC: Pedestrian is Crossing the street, PAC: Pedestrian is About to Cross the street; PA: Pedestrian intention is Ambiguous (PA) 51
2.4 Pedestrian detection using pedestrian actions and id. e.g. pedestrian1 cross= P1C, pedestrian2 preparing to cross=P2PC, pedestrian3 ambiguous=P3A 52
2.5 The InCML Pedestrian Detection Architecture 52
2.6 The JAAD Depth image modality sample 53
2.7 The JAAD Optical Flow image modality sample 54
2.8 Timeline of events/actions whenever the pedestrian is going to cross the street. This image was picked from JAAD [KRT16] dataset source and modified/updated to our requirements. 56
2.9 The Uni-modal true positive detection performance on each image modalities. 59
2.10 The Incremental Cross Modal true positive detection performance. . . 60
2.11 Pedestrian Actions detection performance using ids. The horizontal values represent the pedestrian ids. The Vertical values represent the AP performances. PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC=Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous. 62
2.12 Comparison of Uni Modal Pedestrian Detection and Incremental Cross-Modality Pedestrian Detection. 64
2.13 Comparison of Pedestrian Action Detection for each Imaging modality. 65
2.14 Pedestrian Action Detection performance using ids. The horizontal values represent the pedestrian ids. The Vertical values represent the AP performances. PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC=Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous. 67
2.15 Comparison of Predicted Objects Actions using Uni-Modal Pedestrian Detection and Incremental Pedestrian Detection. PC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC=Pedestrians is About to Cross the street and PA= Pedestrian intention is Ambiguous. 68
2.16 Example of pedestrian actions detection using a different approach. . . 69

3.1 The main architecture of our system. In red are the issues investigated in the first chapter. In blue are the problems studied in second chapter. In orange is the problem analyzed in this chapter 76

3.2 The RNN unrolled architecture 78
3.3 The LSTM representation 79
3.4 The LSTM vs GRU architectures 79
3.5 The RvNN architectures 80
3.6 The RNN based encoder-decoder architectures. 80
3.7 Pedestrian detection using multiple tags 82
3.8 Our time to cross the street estimation method using only BB coordinates in order to estimate the time to cross the street. BB= Bonding Box coordinates, Label= Pedestrian action tag, PPC= Pedestrian Preparing to cross the street; TTC= time to cross; -1= no pedestrian. . . 84
3.9 Our time to cross the street estimation method using the BB coordinates and pedestrian action labels in order to estimate the time to cross the street. BB= Bonding Box coordinates, PPC= Pedestrian is Preparing to Cross the street; PC= Pedestrian is crossing the street; PAC= Pedestrian is About to Cross the street; PA= Pedestrian's intention is Ambiguous; P1,P2= Detected pedestrians; T1,T2= Pedestrian Action Tags; TTC= time to cross; -1= no pedestrian. 85
3.10 A unified CNN-LSTM architecture for detection, recognition and pedestrian action prediction. BBc=Bounding Box Coordinates, P-Tag=Pedestrian action tag, T=Time step, n∈ {1,14}. The CNN has the frames as input data and the LSTM has the pedestrian BBc and pedestrian action tags as input data. 85
3.11 Timeline of event, action and pedestrian behavior whenever the pedestrian is going to cross the street. This image was picked from JAAD [KRT16] dataset source and modified/updated to our requirements. . 87
3.12 The proposed LSTM based architecture for pedestrian time to cross estimation. Input: The BB matrix (4 x 20) at frame T until previews T-n (n= 5, 14, 40), where the X_{i1},Y_{i1},X_{i2},Y_{i2} i=1 to 20 are the BB coordinates for each pedestrian i detected on frame T; output: TTC(i)= number of frames from frame T to the beginning of crossing for the pedestrian i; -1= no pedestrian. 90
3.13 Performance of the time to cross methods using 5 time step. 95
3.14 Performance of the time to cross methods using 14 time step. 96
3.15 Performance of the time to cross methods using 40 time step. 97
3.16 RMNS performance of the time to cross methods using 5 time step. . . 98
3.17 RMNS performance of the time to cross methods using 14 time step. . 99
3.18 RMNS performance of the time to cross methods using 40 time step. . 100

4.1 The main architecture of our upcoming system. 104

A.1 The RTMAPS Detection Architecture for RGB I
A.2 The RTMAPS Detection Architecture for multiple image modality . . . I

List of Tables

1.1 Comparison of learning algorithms and rate policies between AlexNet and LeNet on Caltech dataset 28

1.2 Comparison of learning algorithms and rate policies on Intensity, Depth and Optical Flow Daimler data sets 28

1.3 Comparison of Classical Uni-Modal Learning (UML) vs. Particular Cross-Modality Learning (PaCML) on Non-Occluded Pedestrian Daimler Date Set through LeNet CNN architecture with RMSPROP and POLY Learning Settings 29

1.4 Mean of The Structural Similarity Index (MSSI) on the original image Daimler dataset 30

1.5 Mean of Correlation Coefficient (MR) on the original image Daimler dataset 30

1.6 Mean of the Correlation Coefficient (MR-LOG) on the edge detector images Daimler dataset, using the Laplacian of Gaussian method . . . 31

1.7 Comparison of Correlated (CoCML), Separate (SeCML) vs Incremental Cross-Modality (InCML) Learning Models on the Non-Occluded Daimler Pedestrian Dataset. The results in bold are statistically better than those obtained with the Classical Uni-Modal method. 31

1.8 Optimal Learning Rate and number of iterations for the Incremental Cross Modality Learning with K=10 cross-validation for LeNet and LeNet+ Architectures on Daimler dataset 32

1.9 Single-modality vs. multi-modality on Daimler testing set using images of 36 x 84 pixels. 33

1.10 The performance with late fusion on Non-Occluded Pedestrian Daimler Testing Set. The results in bold are statistically better than those obtained with Classical Uni-Modal Learning. SM=Same Settings, SP=Specific Settings, K-Cross=K-fold Cross-Validation. 35

1.11 The performance with late fusion on Partially Occluded Pedestrian Daimler Testing set. The results in bold are statistically better than those obtained with Classical Uni-Modal Learning. SM=Same Settings, SP=Specific Settings, K-Cross=K-fold Cross-Validation. 35

1.12 Comparison of our models with the state-of-the-art with the false positive rate at 95% True Positive Rate on Daimler dataset 36

1.13 Comparison of AlexNet and VGG-16 with the state-of-the-art on Daimler dataset 37

2.1 Comparison of the pedestrian detection data sets. 54

2.2 Our detection performances using one label. One label represents that all samples are tagged only as a pedestrian (without action recognition). 58
2.3 Our Classical vs InCML detection performances using one label. One label represents that all samples are tagged only as a pedestrian. 60
2.4 The comparison between Faster R-CNN Inception V2 and RetinaNet using the classical unimodal approach on JAAD dataset. The labels represent: PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC= Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous. 61
2.5 Our detection performances using multiple output labels. The labels represent: PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC= Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous. 61
2.6 Our Classical vS InCML detection performances using multiple output labels. The labels represent: PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC= Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous. 63

3.1 The performance of the pedestrian action prediction. ACC BB: Accuracy estimation for the the next bounding box coordinates; RMSE BB: The bounding box Root Mean Square Error; ACC Actions: The actions accuracy estimation for the the next frames; RMSE Actions: The Root Mean Square Error action prediction; ACC Model: The accuracy of the model which is a mean between the ACC BB and ACC Actions; RMSE Model: The Root Mean Square Error is a mean between the RMSE BB and RMSE Actions 92
3.2 The estimation of time to cross method, independently of the detection-classification component. PPC:Pedestrian is Preparing to Cross the street. Real values: testing, independently the pedestrian time to cross estimation on the all real pedestrian samples; Detected Values: testing the detection component connected with the prediction component (time to cross). 93

List of Publications

Articles published in peer-reviewed International Journals (IR)

- **Dănuţ Ovidiu Pop**, Alexandrina Rogozan, Clement Chatelain, Fawzi Nashashibi and Abdelaziz Bensrhair, *Multi-task Deep Learning for Pedestrian Detection, Action Recognition and Time to Cross Prediction*, IEEE Access Jurnal, 2019 (Accepted). Impact factor of 4.098.

- **Dănuţ Ovidiu Pop**, Alexandrina Rogozan, Fawzi Nashashibi and Abdelaziz Bensrhair, *Pedestrian Recognition using Cross-Modality Learning in Convolutional Neural Networks*, Proceedings in IEEE Intelligent Transportation Systems Magazine (ITSM), 2019. Impact factor of 3.294.

  https://hal.inria.fr/hal-02115347/document.

Communications to International Conferences (IC) with reviewing committee and published proceedings

- **Dănuţ Ovidiu Pop**, Alexandrina Rogozan, Fawzi Nashashibi and Abdelaziz Bensrhair, *Fusion of Stereo Vision for Pedestrian Recognition using Convolutional Neural Networks*, In 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 47 -52, April, 2017.

  https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2017-96.pdf

- **Dănuţ Ovidiu Pop**, Alexandrina Rogozan, Fawzi Nashashibi and Abdelaziz Bensrhair, *Incremental Cross-Modality Deep Learning for Pedestrian Recognition*, In 28th IEEE Intelligent Vehicles Symposium (IV), pages 523 -528, June, 2017.

  https://ieeexplore.ieee.org/document/7995771

- **Dănuţ Ovidiu Pop**, Alexandrina Rogozan, Fawzi Nashashibi and Abdelaziz Bensrhair, *Pedestrian Recognition through Different Cross-Modality Deep Learning Methods*, Proceedings of the IEEE International Conference on Vehicular Electronisc and Safety (ICVES), pages 133 – 138 June, 2017. **Nominated for IEEE ICVES 2017 best paper/ best student paper award.**

  https://ieeexplore.ieee.org/document/7991914

- **Dănuţ Ovidiu Pop**, Alexandrina Rogozan, Fawzi Nashashibi and Abdelaziz Bensrhair, *Improving Pedestrian Recognition using Incremental Cross Modality Deep Learning*, In 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 209 -214, April, 2019.

  https://hal.inria.fr/hal-02115319/document

Communications to National Events (CN) with peer review and published proceedings

- **Dănuţ Ovidiu Pop**, Alexandrina Rogozan, Fawzi Nashashibi and Abdelaziz Bensrhair, *Cross Training for Pedestrian recognition using Convolutional Neural networks*, Proceedings in the ORASIS June, 2017.

  https://hal.archives-ouvertes.fr/hal-01866658/document

- **Dănuţ Ovidiu Pop**, *Detection of Pedestrian Actions based on Deep Learning Approach*, Studia UBB Informatica 2019, pages 5-13, 2/2019.

Introduction

This Ph.D. thesis is the result of my research work in the intelligent transportation field to solve the problem of developing a multi-task pedestrian protection system (PPS) including not only pedestrian classification, detection and tracking, but also pedestrian action-unit classification and prediction, and finally pedestrian risk estimation. Moreover, our PPS system uses original cross-modality deep learning approaches.

The pedestrian protection issue is one of the major research directions in the domain of object recognition, computer vision, robotics, and intelligent transportation systems. According to European Commission statistics published in 2016, the number of pedestrians injured in road accidents in 2014 was 1,419,800, and there were 25,900 fatalities.

The number of vehicles on the road has greatly increased over the last few decades. As a consequence, the number of car accidents has also risen, and along with that has grown the need to develop better traffic safety mechanisms.

Traffic safety has become a priority for both the automobile industry and the scientific community, which have invested in the development of different protection systems. Initially, improvements involved simple mechanisms for driver protection such as seat belts, but then more complex systems like anti-lock braking systems (ABS), electronic stabilization programs (ESP) and airbags were developed. Over the last decade, the focus has moved to intelligent on-board systems called Advanced Driver Assistance Systems (ADAS). In the framework of smart vehicles, these systems have to perceive the road environment, to detect road obstacles, to classify their type (cars, cycles, pedestrians) in order to be able to assist the driver and even to stop the vehicle to prevent imminent accidents.

Advanced Driver Assistance Systems (ADAS) could be defined as an intelligent safety system that could enhance the driving experience and ensure better road actor safety. The ADAS system could be set with several options, from simple adaptive cruise control up to a fully autonomous vehicle control. A full ADAS system should include: adaptive light control, adaptive cruise control, hill descent controller, tire pressure monitoring, blind spot detection, automatic parking, intelligent speed adaptation, advanced braking systems, driver drowsiness detection, lane departure warning systems, night vision improvement, road obstacle detection, and pedestrian protection functionality including the estimate and potential prevent a crash or mitigate the severity of a traffic collision. Currently, the academic research and industrial developments propose ADAS systems that include only a part of those functionalities and very few of them address the problem of pedestrian risk estimation. The goal of this thesis is to propose a real time, efficient and robust solution for a multi-task pedestrian protection system able to discriminate between pedestrians and other road obstacles, to identify the pedestrian action units and to estimate pedestrian risk situations.

In the PPS field there are several innovative devices, like the pedestrian detection provided by Bosh which consists of a camera and a radar sensor which warn the driver or automatically launch emergency braking if the system identifies a dangerous situation for pedestrians. The DENSO company has developed a pedestrian detection sensor for the car hood. This sensor aims to decrease pedestrian head injuries in a car collision by creating a larger buffer space between the car hood and hard car components under the hood. The Mobileye detection device consists of a mono-camera automotive pedestrian detection system which can detect pedestrians and cyclists up to a 30-meter range. This system emits audible and visual warnings to alert the driver when there could be a crucial situation. The Delphi detection system has multiple safety functions. It consists of a high camera which allows the system to detect and classify various targets including lane pedestrian, tracking and also inform the driver with a collision warning in a critical situation. A pedestrian and cyclist detection system employing thermal imaging cameras has been developed by Flip company. The thermal imaging camera can make a sharp difference between pedestrians, cyclists, and cars due to its functions to create a clear image based on the temperature stamps of different road users, making it possible to discriminate between them.

Automotive companies like BMW, Mercedes, Nissan, Audi, Toyota, and Peugeot have ADAS technology in the majority of their high-end automobiles. Since 2013, BMW vehicles have been provided with a driver assistance package for pedestrian detection warning. The Mercedes system has connected stereo vision cameras with long, medium, and short-range radars to monitor the area in front of the vehicle. In 2016 the Volkswagen Tiguan was equipped with an advanced radar sensor capable of detecting pedestrians and objects, at a range of up to 170 meters. The Nissan company has developed a system which recognizes the vehicle's environment, including pedestrians, other vehicles, and the road. Lexus RX 2017 has a self-driving system which is linked up to a pedestrian detection system. The Audi ADAS system collects the data from the camera and/or radar sensor to estimate the possibility of a collision by detecting pedestrians or cyclists and alerts the driver with visual, acoustic and haptic warnings if a collision is imminent. The new Peugeot 308 comes with an active safety brake and distance alert. This system detects pedestrians in the car's path and warns the driver if there is a risk of collision. If the driver does not show any reaction in a critical situation or takes too long to, the system immediately activates the automatic brake after a warning. To our knowledge, these ADAS systems do not properly estimate the pedestrian risk situation but only emit a warning whenever a pedestrian is detected within a specific distance range. Our objective is to design an intelligent system able from a previous detected-pedestrian action-unit classification to estimate the pedestrian risk situation.

Moreover, there are some critical aspects that an ADAS should fulfill in order to become a feasible solution that could be implemented on-board a vehicle:

- **the system cost:** this should be reasonably low since the system has to be incorporated in every model of a series;
- **the real-time request:** it has to be fast enough to detect and recognize obstacles in real time, as an obstacle may quickly appear in front of the car and result in an accident. Such a situation is to be avoided at all costs;
- **efficiency:** it should be able to deal with pedestrian occlusions and variable obstacle/ pedestrian shapes and appearances;

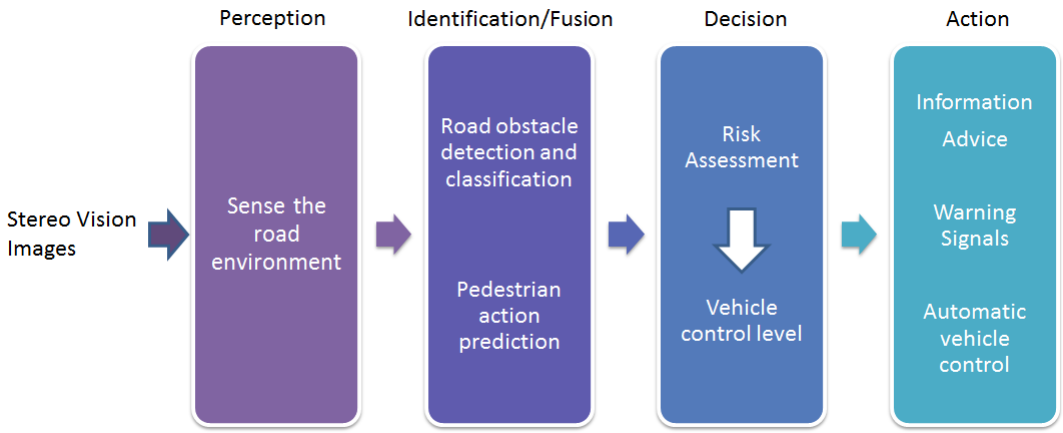


Figure 1 – The main architecture of our system.

- **robustness:** it should work well even in difficult lighting conditions and with a cluttered background;
- **ergonomy:** in order to have a real impact on driver habits, an ADAS has to be ergonomic (especially when drivers are old and/or have physical disabilities), adapted to driver's requirements (asking for warning, assistance or automatic control) and able to take into account the driver behaviour (also hypo-vigilance and emotional states, not only driver actions).

In recent years, many ADAS have been proposed, perceiving the road environment with active sensors (radar, laser scanner, radar, lidar, and sonar) and/or passive sensors (cameras using the visible or infra-red spectrum), without solving all of these problems. The main advantages of using active sensors are their possibility to measure the distance and the speed of the targets and the fact that they work well even in bad weather or poor illumination conditions. However, other issues remain: interference problems, difficulties in interpreting the output signals returned by these sensors for obstacle-classification purposes and the acquisition price, which is usually very high compared to that of vision sensors (the low-cost Valeo's SCALA laser scanner is a refreshing exception). Thus, ADAS employing current technologies are efficient and robust, but they are too expensive, whereas those using only passive sensors are quite cheap, but their functioning still needs to be improved since they struggle in the presence of occlusions and severe lighting conditions.

The goal of our research is to develop an intelligent pedestrian protection component based only on a single stereo vision system using an optimal cross-modality deep learning architecture in order to fulfill the prior requirements.

Our system involves four components (see Figure 1):

1. **The Perception** module senses the road environment with external cameras (monocular or stereo vision) in order to deliver environmental information, including road trajectory estimation, illumination and weather condition detection, obstacle hypothesis generation, among others. This module has been developed by our teams: the STI team at LITIS and the RITS team at Inria Paris.
2. **The Identification/Fusion** module has to detect all road obstacles and among them to identify the most vulnerable ones (i.e. pedestrians), by choosing the

best perception channel or by fusing multi-modal, multi-domain perception channels, according to the environmental conditions and the camera/sensor states. Moreover, this component has to identify the pedestrian action for each frame (crossing, not crossing, and ambiguous action, among others) and to predict not only the pedestrian trajectory but also its action. We study the short, medium and long term prediction approaches. In this thesis, we use the Intensity, Depth and Optical Flow images modality.

3. **The Decision** module has to estimate the risk in order to identify the appropriate vehicle control level (information/advice).

4. **The Action** module has to warn the driver if there is a risk of collision and if necessary the system immediately activates the automatic vehicle control after the warning. The module has been developed by our RITS team at Inria Paris.

The system has to be able not only to detect all the pedestrians with high precision but also to track all the pedestrian paths, to classify the current pedestrian action and to predict their next actions and finally to estimate the pedestrian risk by the time to cross for each pedestrian.

The question is, could we investigate an end-to-end method for the pedestrian detection, classification and prediction components or we must use a sequential method to investigate them separately step by step, according to the target application, as existing approaches from the literature? Our thesis sets out to answer this question following the next methodology: First, we investigate the classification component where we analyzed how learning representations from one modality would enable recognition for other modalitie(s) in various deep learning approaches, which is termed cross-modality learning.

Second, we study how cross modality learning improves an end-to-end pedestrian action detection.

Third, we analyze the pedestrian action prediction and the estimation of pedestrian time to cross.

The thesis is organized as follows:

- **Chapter 1** describes the architecture of our pedestrian classifier and the methods we proposed based on the Cross-Modality deep learning of CNNs based on Daimler [EESG10] and Caltech [DWSP09] datasets.

  The late fusion scheme connected with CNN learning is deeply investigated for pedestrian recognition based on the Daimler stereo vision dataset. Thus, an independent CNN for each imaging modality (Intensity, Depth, and Optical Flow) is used before the fusion of the CNN's probabilistic output scores with a Multi-Layer Perceptron which provides the recognition decision.

  We propose four different learning patterns based on Cross-Modality deep learning of Convolutional Neural Networks:

  1. a Particular Cross-Modality Learning;
  2. a Separate Cross-Modality Learning;
  3. a Correlated Cross-Modality Learning;
  4. an Incremental Cross-Modality Learning model.

Moreover, we also design a new CNN architecture, called LeNet+, which improves the classification performance, not only for each modality classifier, but also for the multi-modality late-fusion scheme. Finally, we propose to learn the LeNet+ model with the incremental cross-modality approach using optimal learning settings, obtained with a K-fold Cross Validation pattern.

This method outperforms the state-of-the-art classifier provided with Daimler datasets on both non-occluded and partially-occluded pedestrian tasks.

- **Chapter 2** is concerned with the pedestrian detection component and action recognition.

  In this chapter, we focus on both pedestrian detection and pedestrian action recognition based on the Joint Attention for Autonomous Driving (JAAD) [KRT16] dataset, applying deep learning approaches.

  The main objective of this approach is to find out if a pedestrian is crossing, or whether the pedestrian's action does not present a critical situation. The most crucial case for the pedestrian and drivers is when the pedestrian is crossing the street in the front of the vehicle, and the car cannot stop or avoid it on time.

  We introduce a unified pedestrian detection component based on deep learning, that also recognizes different pedestrian actions; this is in contrast to usual pedestrian detection methods, which only discriminate between pedestrians and non-pedestrians among other road users.

  We define four main pedestrian actions in order to find out if the pedestrian's action presents a risky situation:

  1. the pedestrian is preparing to cross the street;
  2. the pedestrian is crossing the street;
  3. the pedestrian is about to cross the street;
  4. the pedestrian's intention is ambiguous.

- **Chapter 3** describes the pedestrian detection component merged with the pedestrian action prediction and estimation of time to crossing.

  The pedestrian detection system is one of the vital components of the advanced driver assistance system because it contributes to road safety. The security of the traffic participant could be significantly improved if this system could recognize and predict pedestrian actions or even estimate the time to cross for each pedestrian.

  In this chapter, we focus on pedestrian action prediction, and estimate the time to crossing for each pedestrian. We based this work on the Joint Attention for Autonomous Driving (JAAD) [KRT16] dataset, applying deep learning approaches.

  We propose:

  1. a prediction of pedestrian action using a recurrent deep learning network in order to predict the pedestrian's next actions on the short (T+1, T+2, T+3, T+4, T+5), medium (T+14) and long time (T+40);

2. an estimation of time to cross for a single and multiple pedestrians using recurrent deep learning network.

We use an Long Short-Term Memory (LSTM) [HS97] to estimate the pedestrian intention action using the previous 5, 14 and respectively 40 frames as time steps. We show that integrating multiple pedestrian tags for the detection part, merged with LSTM, can achieve a significant performance.

- Finally, in **Chapter 4** we present our conclusions and discuss future work.

Chapter 1

Cross-Modality Pedestrian Recognition

Contents

1.1 Introduction **11**

1.2 Related Work **14**

1.2.1 Handcrafted Features Models 14

1.2.2 Deep Learning Features Models 15

1.2.3 Handcrafted Features Models vs. Deep Learning Features Models 16

1.3 Proposed Architectures for Pedestrian Classier **17**

1.3.1 Classical Learning Approach 19

1.3.2 LeNet+ Convolutional Neural Network Architecture 20

1.3.3 Early Fusion architecture 21

1.3.4 Late Fusion architecture 21

1.3.5 Cross-Modality Learning Approaches 22

1.3.6 Late Fusion Pedestrian Classification with Incremental Cross-Modality Learning 25

1.4 Evaluation of Classification Components **26**

1.4.1 Experimental Setups and Evaluation Protocol 26

1.4.2 Evaluation of Uni-Modal Learning Classifiers 27

1.4.3 Evaluation of the Particular Cross-Modality Learning Classifier 28

1.4.4 Comparison of Uni-Modal Classifiers with Cross-Modality Learning Models 29

1.4.5 Early-Fusion vs Late-Fusion with Classical Learning method 33

1.4.6 Late-Fusion with Classical vs Cross-Modality Learning using LeNet and LeNet+ CNN architectures 34

1.4.7 Late-Fusion with Classical vs Incremental Cross-Modality Learning using AlexNet and VGG-16 CNN architectures 35

1.4.8 Comparisons with the state-of-the-art methods 36

1.5 Conclusion **39**

1.1 Introduction

Pedestrian detection is a highly debated issue in the scientific community due to its major importance for a large number of applications, especially in the fields of automotive safety, robotics and surveillance. In spite of the widely varying methods developed in recent years, pedestrian detection is still an open challenge whose accuracy and robustness has to be improved.

A pedestrian detection system has three main components: the sensors used to capture the visual data, the modality image processing components and the classification components. In general, all these components are processed and developed together to obtain a high detection performance, but sometimes each element could be investigated separately according to the target application. The main difference between the detection and classification is that the classification task is the process of distinguishing the objects/images between them and classifying the objects/images to some categories, based on specifics features, while the detection task is the process of finding out the particular objects in the images, that involves both classification and localization. Therefore the classification requires features information from detection task.

This Chapter is concerned with improving the classification task, which is the central part of the pedestrian detector.

In recent years, deep learning classification methods, in particular Convolutional Neural Networks (CNNs), combined with multi-modality images applied on different fusion schemes have achieved great performances in computer vision tasks. For the pedestrian recognition task, the late-fusion scheme outperforms the early and intermediate integration of modalities.

These CNNs differ in size and depth according to the objects that need to be classified. Thus, with an increase in the complexity of the classifier's problem, the CNN's size and depth also increase, which usually enhances the CNN's performance.

The drawback of CNNs with very large and complex architectures, such as GoogleNet, VGG, is that they require considerable computing power and a vast storage space, especially for the off-line learning process, but also to a lesser degree for the on-line classification applications.

This problem has been partially solved since for the off-line step the CNNs could be learnt on an expensive powerful network of computers, but it could be an unsolved problem for several on-line embedded applications. Indeed, the CNNs involved in an ADAS system should fulfill some requirements to become a feasible solution for on-board implementation in a vehicle.

One of them is the system cost. It should be low enough since it has to be incorporated in every series vehicles. We believe that those series vehicles with ADAS system incorporate must be affordable for all population categories.

The question is: could we adapt a vast CNN architecture to be a feasible solution in order to upload it into a cheap embedded processing module or should we create a new one to fit on the required ADAS settings? We propose to investigate this question in the section 1.3.2 were we designed a new CNN. Increasing the CNN's complexity (architecture and learning settings), the classifier models require higher computing power for off-line learning and on-line applications that lead to the purchase of more powerful and expensive GPUs.

In order to solve this issue we can increase the hardware capabilities which are in continuously evolving, which today it makes sense to use more complex network

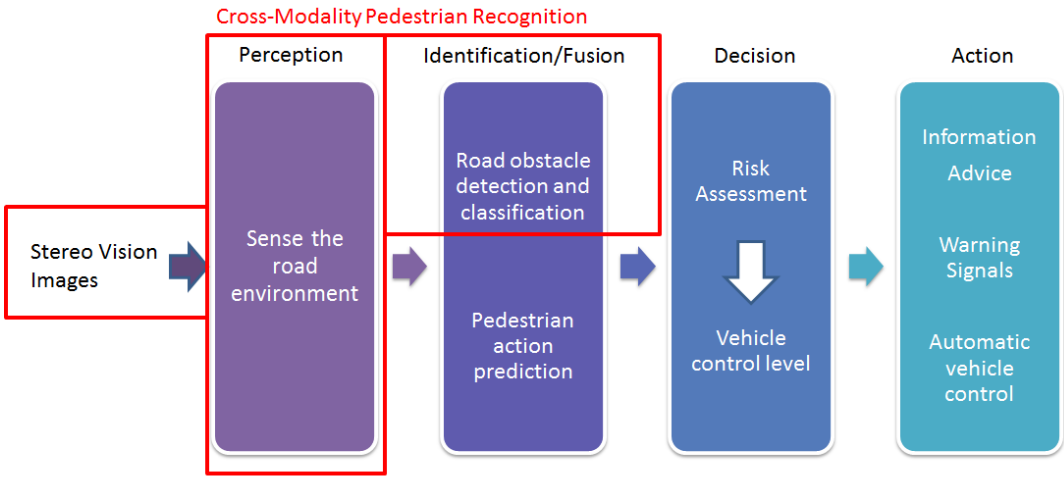


Figure 1.1 – The main architecture of our system. In red are the issues investigated in this Chapter.

architectures in ADAS applications (as long as a proper region proposal method is used), or to adapt a CNN architectures which should be compact enough to allow an efficient and real-time implementation, even on a cheap Nvidia embedded platform with limited memory size and quite sluggish processing time, to make their use possible in every series vehicles.

We chose to propose a compact, but efficient CNN architecture for the pedestrian recognition task, one that is well-suited to small-size multi-modal images derived from stereo vision.

Deep learning classification methods associated with multi-modality images and different fusion patterns have achieved notable performances in many applications, including pedestrian classification issue.

This Chapter investigates how a multi-modal system could be learnt when data in one of the modalities is scarce (e.g. many more images in the visual spectrum than depth). If the system is learnt on multi-modal data, could it still work when the data from one of the domains is missing? Could the learning process be improved if it uses a different image modality validation set than the training set?

This Chapter sets out to evaluate this cross-modality concept through various experiments based on the Daimler stereo vision dataset [EESG10] and will allow us to chose the most promising one for this pedestrian classification task.

The main contribution of this chapter is concerned with investigating different cross-modality learning approaches for deep neural networks aimed at the pedestrian recognition task on both non-occluded and partially-occluded samples, using various sensor modalities. It also proposes a new variation of the LeNet architecture and provides results for a late-fusion approach. Thus according to our main thesis object described in the introduction (see Figure 1), in this chapter, we analyze only the first two components (see Figure 1.1):

- Perception: using the stereo vision images;
- Identification/Fusion: pedestrian classification using cross-modality deep learning approach.

This Chapter is organized as follows:

- Subsection 1.2 shows the related work from the literature.
- Subsection 1.3 presents the classification architecture and the associated learning methods based on Cross-Modality deep learning of CNNs.
- Subsection 1.4 describes the evaluation protocol in order to asses our classification approaches.
- Finally, Subsection 1.5 presents our conclusion for this chapter.

1.2 Related Work

The pedestrian classification issue has attracted considerable interest over the last decade, resulting in a wide variety of detection methods. This issue has been widely investigated, but it still remains an open challenge because the detection progress is hindered by the difficulty of detecting all partially occluded pedestrians and the problem of classifying in severe weather conditions.

This subsection presents the background knowledge related to the pedestrian classification system. It presents recent scientific investigations, which can generally be classified in two categories: Handcrafted Features Models and Deep Learning Neural Network Models.

1.2.1 Handcrafted Features Models

Pedestrian classification is one of the most significant issues in computer vision research and object recognition. Over the last decade, this problem has been more deeply investigated through the development of classification methods using a combination of features such as Integral Channel Features [DTPB09], Histograms of Oriented Gradients (HOG) [DT05b], Local Binary Patterns (LBP), Scale Invariant Feature Transform (SIFT) [VGVZ09a], among others [SKHD09, FGMR10a], followed by a trainable classifier such as a Support Vector Machine (SVM) [FGMR10a], Multi-Layer Perceptron (MLP), boosted classifiers [DTPB09] and random forests [BOHS14, DWSP12a].

A comprehensive review based on 16 state-of-the-art detectors over 6 data sets made up to 2012 is presented in [DWSP12b], where it brought together and studied a annotated pedestrian detection including size, occlusion and pedestrian positions in public scenes.

A detailed investigation of 40+ detectors made up to 2014, based on the Caltech pedestrian detection benchmark is introduced in [BOHS14]. The paper presents the most promising ideas from multiple published strategies and a comparison of the contemporary pedestrian datasets.

A recent strengthening of results for pedestrian detection using HOG, LUV, and optical flow as features with the AdaBoost classifier based on Caltech-USA pedestrian dataset is presented in [RSZS16]. An improved fast multiscale pedestrian detection algorithm based on integral channel features and cascade AdaBoost classifiers is presented in [GHW18]. The RGB video is converted into LUV images, and then the image pyramid is determined to obtain the channel features. A trilinear interpolation in HOG feature merged with SVM is introduced in [PP17], where it is applied to two planes of training, the learnt classifier and the full-body classifier with the estimated scores. A Haar wavelet decomposition and HOG feature extraction with a basic statistical operator for adapting to a binary classification based on a Support Vector Regression (SVR) is presented in [ER16]. This method is applied on a public pedestrian dataset and compared with K-Nearest Neighbors (KNN) and SVM classifiers.

We chose to present the state-of-the-art models given with the Daimler data sets, since our classification models are developed on these datasets.

In so far as we are going to work with multiple image modality, we selected to employ the Daimler dataset in our preliminary pedestrian classification experiments because the authors of the dataset have already defined the Intensity, Depth,

and Optical Flow image modalities including the learning and testing samples, and that enable us to perform a fair comparison benchmark.

A Mixture-of-Experts (MoE) framework performed with HOG and LBP features, and MLP or linear SVM classifiers was presented in [EESG10, EG11]. In the HOG/ linSVM MoE, the HOG descriptor was computed with 12 orientation bins and 6 x 6 pixel cells, accumulated for overlapping 12 x 12 pixel blocks with a spatial shift of 6 pixels, and then those features were inserted into linerar SVM [EESG10]. In the HOG+LBP/ MLP Mixture-of-Experts (MoE), the HOG and LBP features were inserted into MLP [EG11]. The HOG descriptor was applied with 9 orientation bins and 8 x 8 pixels cells, accumulated for overlapping 16 x 16 pixels blocks with a spatial shift of 8 pixels. The LBP descriptor was applied using 8 x 8 pixel cells and a maximum number of 0-1 transitions of 2. Those feature-based Mixture-of-Experts (MoE) models are learnt using a classical learning methodology where both learning and testing were done on the same modality: Intensity, Depth or Optical Flow.

1.2.2 Deep Learning Features Models

In recent research studies, deep learning neural networks including convolutional neural networks (CNNs), like LeNet [LBBH98], VGG [SZ14], GoogLeNet [SLJ+14], have usually led to improvement in classification performance [HOBS15, FYY+15].

A deformation part-based model combined with a deep model based on a restricted Boltzmann Machine for pedestrian detection is presented in [OW12]. The deformation-part component receives the scores of pedestrian body-part detectors and provides a decision hypothesis to the deep model in order to discriminate the visibility correlation among overlapping elements at multilayers. This approach was applied not only on the Daimler datasets but also on the Caltech, ETH and CUHK datasets. A deep unified model that conjointly learns feature extraction, deformation handling, occlusion handling and classification evaluated on the Caltech and ETH datasets for pedestrian detection was proposed in [OZL+17]. A solution for detecting pedestrians at different scales and evaluated on the Caltech dataset by combining three CNNs was proposed in [ESWG16]. A cascade Aggregated Channel Features detector is used in [XWK+15] to create pedestrian candidate windows followed by a CNN-based classifier for assessment purposes on monocular Caltech and stereo ETH data sets.

Recently, in [BDX16] a CNN to learn the features with an end-to-end approach was presented. This experiment focused on the detection of small scale pedestrians on the Caltech dataset.

A pedestrian detection system based on the Gabor filter, HOG and CNN employing the INRIA and Daimler Mono Pedestrian dataset is presented in [ATI19] where both data sets were applied for training, testing and the PennFidanPed dataset only for testing.

The visual contexts based on scale and occlusion cues from detection at proximity using CNN for a better pedestrian detection for surveillance applications was introduced in [WXY16].

In [YML17], the authors presented a new multi-scale classifier method based on CNN, using the nearby scale classifier instead of extracting features multiple times from the resizing images and also introduced a Binary Pattern of Gradient (BPG) in order to accelerate the feature extraction speed.

Two CNN-based fusion methods (early and intermediate fusion architectures)

Figure 1.2 – The Handcrafted Features Model Architecture. The ROI Processing represents the regions of interest from the proposed image, which inserts it in a feature extraction algorithm and then inserts this emphasizes information in the learning classifier process. Whenever the learning process is over; it returns the trained classifier, which could be used in the real-time application.

of thermal and visible images were presented in [WFHB16] and evaluated on the KAIST pedestrian dataset. The early fusion approach merges the information of these modalities at the pixel level, the intermediate fusion method generates a feature representation for each modality using separate sub-networks before classification. The authors concluded that intermediate fusion has greater classification accuracy than early fusion.

In the literature, for the late fusion architectures, the learning is performed independently on each modality, with annotated images provided exclusively from that modality. To the best of our knowledge, no study has been carried out on cross-modality learning for pedestrian recognition, but only on cross-dataset learning. Thus, in [KG16], the authors proposed an incremental cross-dataset learning algorithm for the pedestrian detection problem. A synthetic dataset (Virtual Pedestrian dataset [VLM+14]) is used for basic training and two distinct real-world datasets (KITTI Vision Benchmark Suite and the Daimler Mono Pedestrian Detection Benchmark) for fine-tuning the models and for evaluation.

1.2.3 Handcrafted Features Models vs. Deep Learning Features Models

The handcrafted feature is usually handled with classical machine learning methods for object recognition and computer vision. It address to features determined using several algorithms employing the information present in the image like edges and corners.

The handcrafted features methods mentioned above usually are processed in three steps [AH16, NGB17]: a detection algorithm which locates the regions of interest from the proposed image, then a feature algorithm which extracts the characteristics information and then a classification algorithm which distinguishing the designated area on each particular features from the others (see Figure 1.2). The set of features considered for creating the descriptor depends on the specific feature being used. The handling proceeding of handcrafted features represent a drawback because of processing time and often involves finding the right trade-off between accuracy and computational efficiency. On the other hand, some of most of them has been solved the shortcomings issue in the speed and efficiency and were proposed to be applied in real time applications on devices with low computational power.

The Deep Learning progress exceeded the image classification issue [KSH12a] using handcraft features. It usually consists in complex networks for solving the problem of image classification, regularly addressed through CNNs, where deep lay-

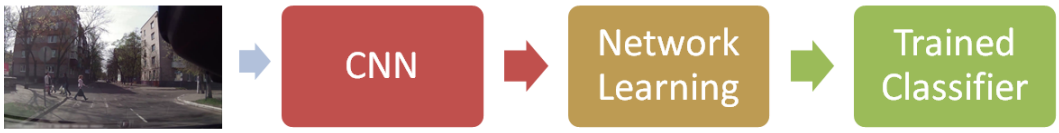


Figure 1.3 – The Deep Learning Features Model Architecture. The CNN extract implicit the features from the image which are learnt in the same step. When the learning stage is completed, it returns the trained classifier, which could be used in the real-time application.

ers in these complex networks perform as a collection of feature extractors that are commonly quite generic and somehow independent in the particular classification task [Sch15,LBH15] (see Figure 1.3. The idea of this method is to create multiple levels of representation so that higher-level features can describe the meaning of the data, which in turn can render higher robustness to intra-class variability [CJG+15].

Therefore, it is reasonable to consider the deep layers of CNN as a feature extractor. The significant difference consists in that the features extracted by a CNN are learned to employ the data in contrast to hand-crafted features, that must be created before by researchers to achieve a given set of chosen characteristics.

Since the Deep Learning methods achieve better accuracy than Handcrafted feature approaches, we decided to use this technique in our research.

The drawback of CNN models is that they need a large amount of annotated data for each modality. It usually happens that one has not (enough) annotated data in one modality compared with other modalities.

The question is whether one modality can be used exclusively (standpoint one) for training the classification model used to recognize pedestrians in another modality or only partially (standpoint two) for improving the training of the classification model in another modality. To our knowledge, these questions have not yet been answered for the pedestrian recognition task, the fact that we decided to merge the cross-modality method with deep learning strategy.

1.3 Proposed Architectures for Pedestrian Classier

In this section, we present our proposed cross-modality pedestrian classification architectures, including explanations and setups of our approaches.

We believe it is necessary to improve the classification component of an ADAS system to be able to discriminate between the obstacle type (pedestrian, cyclist, child, old person) in order to adapt the car driver systems behavior according to the estimated level of risk. This Chapter is concerned with improving the classification component of a pedestrian detector. The work presented in this chapter aims to train a CNN-based classifier using a combination of Intensity, Depth and Optical Flow modalities on the Daimler [EESG10] stereo vision dataset.

To achieve this aim, we develop the classification component based on four CNNs:

1. Lenet [LBBH98] as it is a straightforward and small architecture which allows better running even on a CPU (using small image size, the default is 32x32 pixels);

2. Lenet+ which we proposed, is a variation of Lenet and improves the classification performance for each modality classifier;

3. AlexNet [KSH12a] for its incontestable impact on machine learning due to a good balance between its performance and compact architecture;

4. VGG-16 [SZ14] because of its high performance obtained with such a vast architecture commonly used in pedestrian detection.

To do so, we followed the procedure below, relying on a deep learning approach:

- Investigating the performances of AlexNet [KSH12a] and LeNet [LBBH98] on the Caltech [DWSP09] dataset using RGB image modality where pedestrian bounding boxes (BBs) are more than 50 pixels. All BB were resized to quadratic size (64 x 64 pixels) to obtain a better performance.

- Evaluating the LeNet architecture with various learning algorithms and learning rate policies using the classical learning method for each Intensity, Depth and Optical Flow image modalities;

- Combining three image modalities (intensity, depth and optical flow) to feed a unique Convolutional Neural Network (CNN), using an Early fusion method and fusing the results of three independent CNNs, using Late fusion method;

- Evaluating a Particular Cross-Modality learning method where a CNN is trained and validated on the same image modality, but tested on a different one;

- Evaluating a Separate Cross-Modality learning method which uses a different image modality for training than for validation;

- Evaluating a Correlated Cross-Modality learning method where a unique CNN is learnt (trained and validated) with Intensity, Depth and respectively Optical Flow images for each frame;

- Evaluating an Incremental Cross-Modality learning where a CNN is learnt with the first images modality frames, then a second CNN, initialized by transfer learning on the first CNN, is learnt on the second image modality frames, and finally a third CNN initialized on the second CNN, is learnt on the last image modality frames;

- Improving the incremental cross-modality learning due to a new CNN (we called Lenet+) architecture that we proposed together with K-fold Cross Validation of both the learning rate and epoch numbers;

- Learning on AlexNet [KSH12a] and VGG-16 [SZ14] using the default CNNs setting with the Classical Learning method and respectively with the Incremental Cross Modality Deep Learning method on Intensity, Depth and Optical Flow image modality;

- Optimizing the CNNs hyper-parameters (convolution stride, kernel size, convolution number of outputs, weights of the fully connected layers) for the classical learning method and for the incremental cross modality deep learning method respectively;

- Implementing the late fusion scheme with Support Vector Machine (SVM) [FGMR10b] for classical learning approach;

- Implementing the late fusion scheme with Multi-Layer Perceptron (MLP) for both classical and incremental cross-modality learning methods considered above.

We benchmark different learning algorithms and rate policies using the LeNet architecture. We show that the late-fusion classifier outperforms not only all single modalities but also the early-fusion classifier.

We examine all these methods with the classical learning one where each CNN is learnt and evaluated on the same image modality. We also compare all these learning patterns with the classical learning approaches within the MoE framework proposed in [EESG10, EG11] and deep Boltzmann-Machine [OW12] for the recognition of both partially occluded and non-occluded pedestrians.

The following subsection describes the architecture and the corresponding settings for each of the classical and respectively cross-modality learning methods.

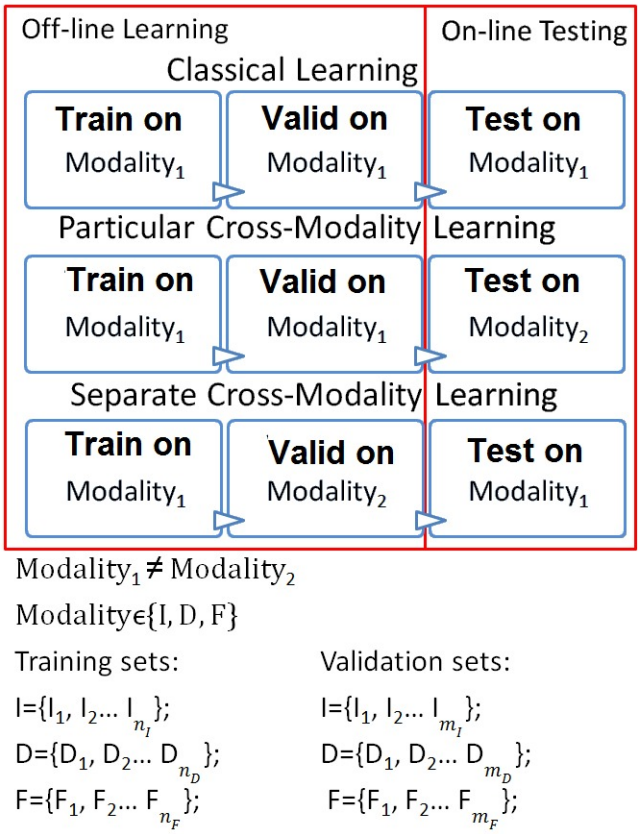


Figure 1.4 – The classical learning approach uses the same image modality for the training, validation, and testing processes. The Particular Cross-Modality learning uses the same image modality for training and validation, but a different one for testing. The Separate Cross-Modality learning uses the same image modality for training and testing, but a different one for validation.

1.3.1 Classical Learning Approach

The Classical Learning (CL) method involves that both training and validating are performed on the same image modality. For each image modality, a classifier model is fitted on the respective training dataset; successively, the fitted model is used to predict the labels for the observations in the validation dataset; and finally, the test

dataset is used to provide an unbiased evaluation of the final model fitted on the learning dataset (union of training and validation datasets). For the classical learning approach, we have trained, validated and evaluated each CNN with the same imaging modality either Intensity, Depth, or Optical Flow (see Fig.1.4).

We start by comparing AlexNet to LeNet with different learning algorithms: Stochastic Gradient Descent (SGD) [Bot12], Adaptive Gradient (ADAGRAD) [DHS10], Nesterov Accelerated Gradient (NAG), RMSPROP, ADAM and learning rate polices: Step Down (STEP), Polynomial Decay (POLY) and Inverse Decay (INV) on the intensity modality. From the Caltech image dataset we selected pedestrians bounding boxes (BB) of more than 50 pixels. All the BB were resized to quadratic size (64 x 64 pixels), that allows to obtain a local minimizer of the quadratic criterion easily and obtains a better performance [LBBH98]. We observed that the LeNet provides better results than AlexNet for these small size image datasets samples.

In the second experiment, we evaluated the LeNet architecture with various learning algorithms on the Daimler [EESG10] dataset: Stochastic Gradient Descent (SGD) [Bot12], Adaptive Gradient [DHS10], RMSPROP [TG12], ADADELTA and learning rate policies: Fixed (FIX), Exponential (EXP) [Sun13], Step Down, Polynomial Decay (POLY) [BT10], Sigmoid, Multi-Step and Inverse Decay. Each modality classifier was exclusively trained with images of its own modality using the original images size (96 x 46 px). We conclude that various modalities require different learning algorithms and rate policies for an efficient learning but an equivalent number of iteration and similar initial learning rate.

1.3.2 LeNet+ Convolutional Neural Network Architecture

Each modality CNN, was first set up on the LeNet architecture [LBBH98]. We observed that the LeNet has a limited generalization power for our needs. In order to enhance the classification performance and avoid overfitting, we designed a CNN, which we called LeNet+, (see Fig 1.5) by extending the LeNet architecture by adding three layers and replacing the weight filler algorithm from FC layers. We add a ReLU layer, a Local Response Normalization (LRN) layer [KSH12b] at the first Pooling Layer, a Dropout layer [SHK+14] with a rate of 50% at the first FC layer. Moreover, for the weight filler we use the Gaussian [MU08] instead of the Xavier algorithm [GB10]. For the FC layers, we used 4096 neurons for the first FC layer and two neurons for the second FC layer.

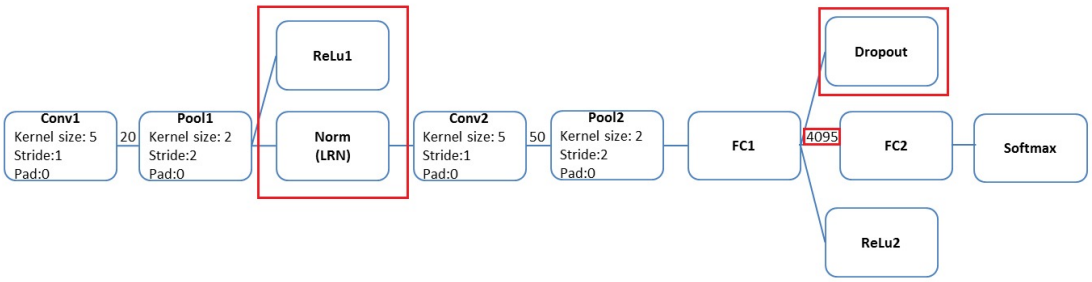


Figure 1.5 – The proposed extended LeNet Architecture (LeNet+). The extension consists in adding a ReLu and an LRN layer at the first Pooling layer, adding a Dropout layer at the first FC layer, using the Gaussian instead of Xavier algorithm for the weight filler and increasing the outputs for the first FC layer from 500 to 4096.

1.3.3 Early Fusion architecture

The early fusion approach integrates three image modalities (Intensity, Depth and Optical Flow) by concatenating them to learn a single CNN (see Figure 1.6).

It is less efficient and robust than the late-fusion model. Thus, the early-fusion model requires high image calibration and synchronization. The early-fusion training method is more constrainable since for a given image frame it needs an item for each modality, and therefore the classifier requires more samples to learn the problem. With the early-fusion model, it is impossible to take advantage of cross-dataset training methods, by using modality images from different uni-modal and/or multi-modal datasets where all the modalities involved are not acquired and/or annotated. The early fusion method does not allow one to improve the learning by extending the number and the variety of items through the cross modality learning. The performance of the early fusion and late fusion models on the Daimler stereo vision dataset were compared in our work (see subsection 1.4.5).

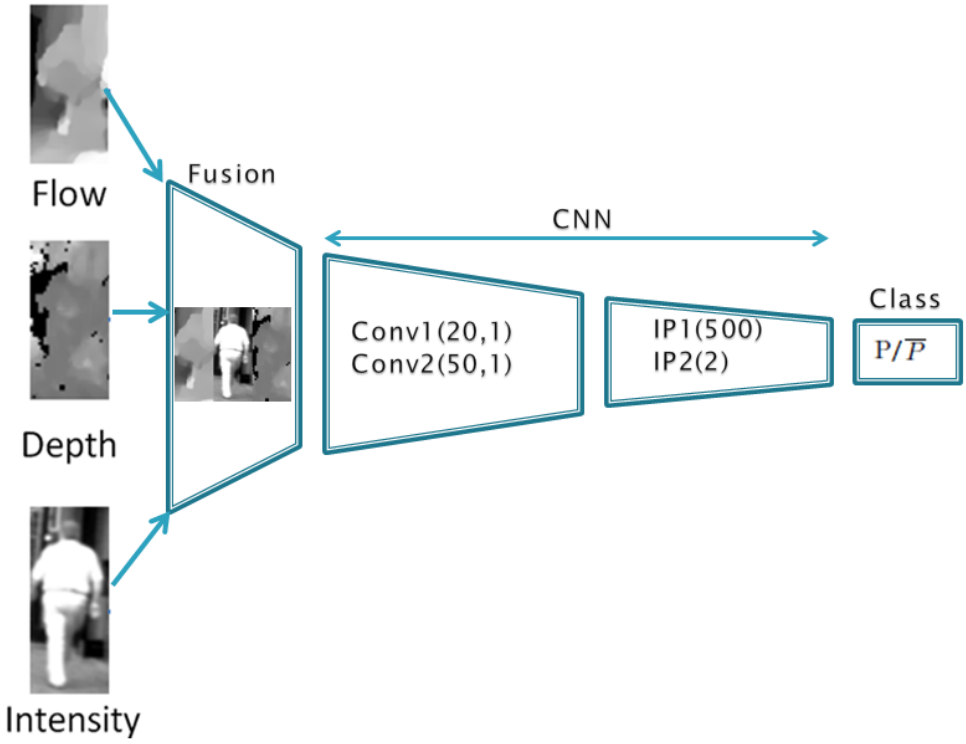


Figure 1.6 – The Early Fusion Architecture

1.3.4 Late Fusion architecture

We proposed a late-fusion architecture using two approaches:

- We propose the fusion of the Intensity, Depth and Flow modalities within a hybrid CNN-SVM framework. We train an SVM to discriminate between pedestrians (P) and non-pedestrians ($\overline{P}$) on the classification results of the three independent CNNs;
- We propose a late-fusion architecture (see Figure 1.7) where an MLP is used to discriminate between pedestrians (P) and non-pedestrians ($\overline{P}$) on the classification results (it combines the output scores of all classifiers) of three modality CNNs.

Each CNN is exclusively trained with images from the same modality (among Intensity, Depth and Optical flow) and then tested on that modality images. All models are learnt and compared with Daimler [EESG10, EG11] dataset.

The final layer for each CNN returns the classifier probabilistic scores from CNN's and after that, an MLP/SVM fuses the classifier probabilistic scores to obtain the final decision of the classifier system: P or $\overline{P}$.

We believe that the late-fusion we propose based on three independent CNNs followed by an SVM or MLP is a promising approach because a sequential learning usually provides better results.

The off-line learning of the late fusion scheme is therefore costly, but it is an efficient solution for on-line applications.

Analyzing the CNN-SVM late fusion model, it did not outperform the best Daimler classifier which is carried out with HOG+LBP/MLP [EESG10, EG11]. We decided to use only the MLP as in [EESG10, EG11] in our next late-fusion experiments, to maintain a fair comparison.

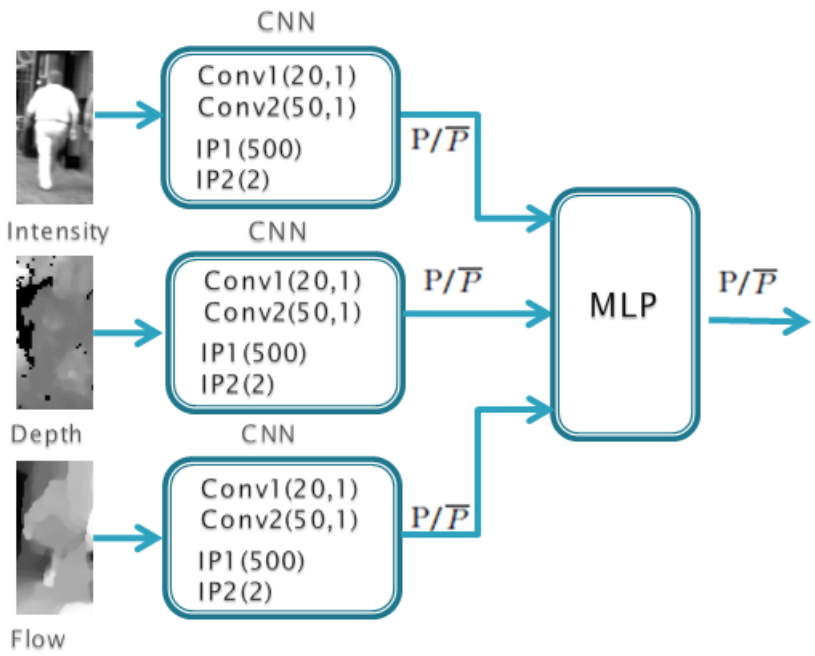


Figure 1.7 – The MLP Late-Fusion Architecture

1.3.5 Cross-Modality Learning Approaches

In the first part, we studied three methods of integrating different image modalities (Intensity, Depth, Optical Flow) to improve pedestrians detection. In the second part, we studied how learning representations from one modality would enable prediction for other modalities, which they term as Cross-Modality Learning.

Thus, the following methods analyzed cross-modality learning through various experiments based on the Daimler stereo vision dataset.

Particular Cross-Modality Learning

We propose a Particular Cross-Modality Learning (PaCML) that carries out the learning process on the same image modality, although the training and validation sets

are disjoint, and the performance is evaluated on a different modality. This approach shows whether the automatic annotation of modality images could be extracted with a classifier trained with different modality data (see Fig.1.4).

Separate Cross-Modality Learning

We also propose a Separate Cross-Modality Learning (SeCML) that carries out the learning process when the modality of the training set differs from that of the validation set. The testing set belongs to the same modality as the training set (see Fig.1.4). This approach could improve the generalization power of CNN and shows how we could train a system when one of the imaging modalities is limited.

Correlated Cross-Modality Learning

Then we design a Correlated Cross-Modality Learning (CoCML) approach that learns a single CNN, where the data training set consists of frames with distinct image modalities: Intensity I_i, Depth D_i and Flow F_i with i=$\overline{1,n}$ (see Fig. 1.8).

The CNN model is validated in two different ways: on a multi-modal validation set (a stack of images from the same frames for different image modalities) and respectively on a uni-modal validation set. The training and validation sets are disjoint.

We consider that the disadvantage of CoCML is that it requires to use an identical CNN model. This weakness is a considerable restriction if distinct modalities improve the learning process with a specific CNN architecture and/or with various settings (i.e., learning rate policies and learning algorithms).

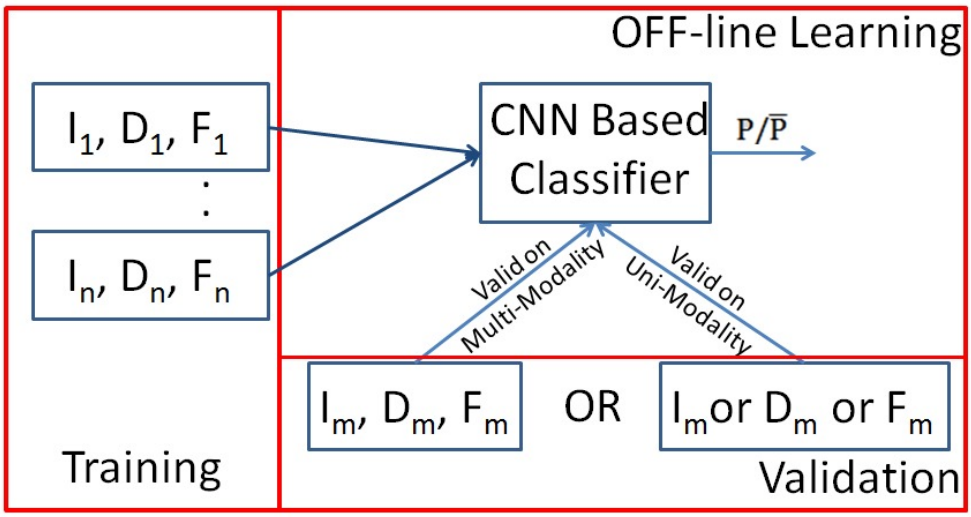


Figure 1.8 – Correlated Cross-Modality Learning. The learning data consists of Multi-Modal Correlated images presented successively to the CNN for training and respectively in Multi-Modal or Uni-Modal images for validation. I∈{I_1,I_2 ... I_n}; D∈{D_1,D_2 ... D_n}; F∈{F_1,F_2 ... F_n}; I=Intensity; D=Depth; F=Optical Flow.

Incremental Cross-Modality Learning

In our incremental cross-modality approach, a first CNN is learned (trained and validated) with the first image modality frames, then a second one, initialized by transfer learning on the primary CNN, is learned on the second image modality frames,

and finally a third one initialized on the second CNN, is learned on the last modality image frames. This model does not require correlated modality frames or equal numbers of items (see Fig.1.9).

This method has some advantages compared with both classical and others cross modality methods. One of the benefits is that it is more flexible than the previous cross-modality learning methods. This method allows different settings to be adapted for each classifier (i.e. different learning rate policies and learning algorithms) which leads to better learning for the final classification system. Transferring the weight information from one classifier which was already learned to another one which will be learnt next, increases the ability of the model to discriminate with a distinct point of view for the same standard target class of modalities (i.e. pedestrians or non pedestrians). It allows additional learning with other modality images without changing the concept target class.

Learning this model does not require any calibration and/or synchronization between modality images. This approach could be adapted and utilized when the multi-modality images are not derived from the same database and/ or obtained from related sensors/ cameras. Moreover, this procedure can be suitable for using various data sets and stretch out in cross-dataset training.

This approach casts doubt upon whether the learning image modality order could affect the performance of the final classifier. We have examined various combinations by interchanging the imaging modalities, and conclude that to classify the Intensity image modality, the training process needs to start with Depth modality, followed by Optical Flow and finally Intensity images (D, F, I training model of I). The optimal learning order for Optical Flow image modality classification is Depth images, followed by Intensity images and finally Flow images (D, I, F training model of F). To achieve the best learning performance for Depth modality, the training process should start with Intensity images followed by Flow images and finally Depth images (I, F, D for the training of D).

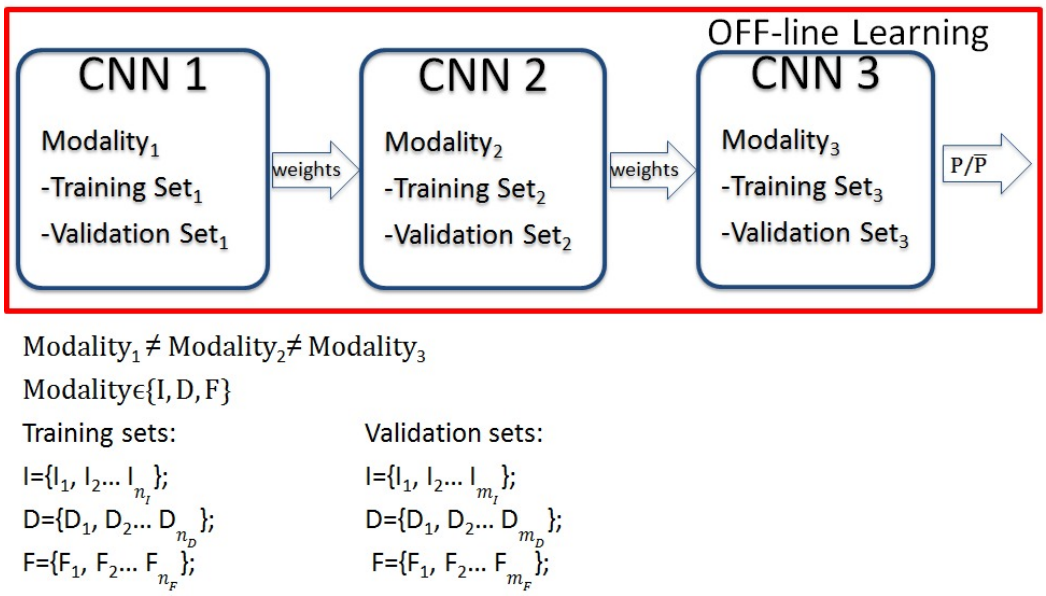


Figure 1.9 – Incremental Cross-Modality Learning. The first CNN is learning (training + validation) on the same image modality. When the learning process is over, the weights information from the previous CNN is transferred to the next CNN in which the learning process starts with a different image modality.

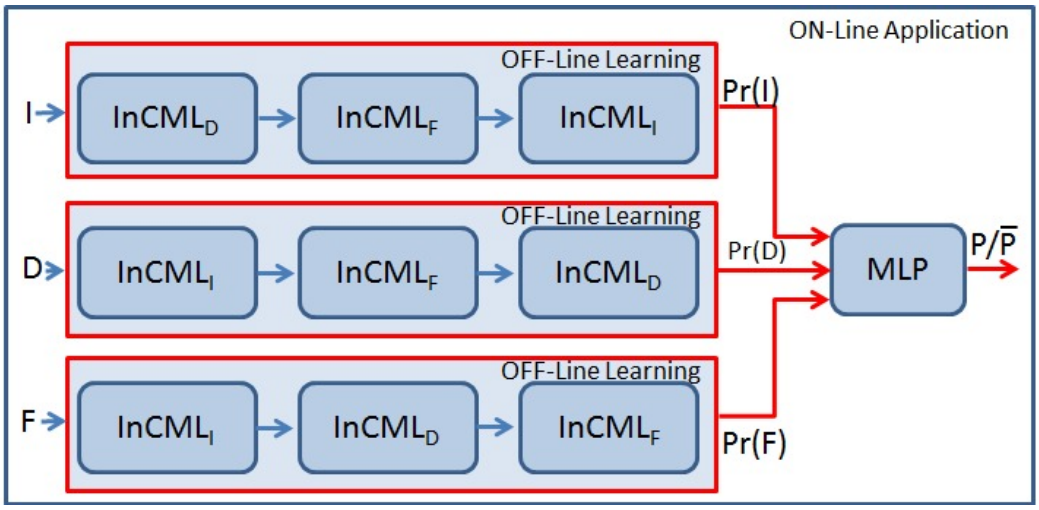


Figure 1.10 – The Late Fusion Architecture with the Incremental Cross-Modality Learning. The modality probabilistic output scores of Intensity (Pr(I)), Depth (Pr(D)) and Optical Flow (Pr(F)).

1.3.6 Late Fusion Pedestrian Classification with Incremental Cross-Modality Learning

Our late-fusion architecture (see Fig 1.10) consists of three independent CNN classifiers and an MLP that discriminates between pedestrians (P) and non-pedestrians ($\overline{P}$) based on class probabilistic estimates provided by each CNN. The learning process for each CNN classifier is done with an incremental cross-modality learning approach in an independent manner. The last layer of each CNN provides only one of the modality probability output scores among Intensity Pr(I), Depth Pr(D) and Optical Flow Pr(F). The MLP is composed of three neurons in the input layer, one hidden layer with 100 neurons, and 2 neurons in the output layer. We used the ReLU function for the activation function and a Stochastic Gradient Descent (SGD) [Bot12] solver for the weight optimization. For the weight updates, we used a constant learning rate (1e-07).

Late fusion focuses on three independent components for the learning of modalities, and then, the probabilistic output scores are fused into a multi-modal representation for the final learning step. The off-line learning of the late fusion scheme is therefore costly but it is an efficient paralelisable solution for on-line applications.

We experimented out target classification task with different CNNs (VGG [SZ14] [LBBH98], AlexNet [KSH12a], GoogleNet [SLJ+14] and LeNet [LBBH98]) and various hyperparameters. Concerning the input image size, the bounding box in Daimler sets are images of 48x96 pixels. We resized the input layer size accordingly to 48x96 pixels for the LeNet, AlexNet, GoogleNet, VGG. The AlexNet and VGG did not return suitable results for this input image size, because these CNNs is designed to achieve better performance with large-scale images, while LeNet is expected to handle even with small-images suitably. To solve this problem the solution is to increase the image input size by interpolation and the best results were obtained with 256x256 pixels for AlexNet, GoogleNet and VGG. However, the performances obtained with those more sophisticated architectures are less than those obtained with LeNet on the Daimler dataset. Another solution to avoid this problem is to remove some layers and/or to change the convolution parameters and num-output options in the

convolution and inner product layers. This is equivalent to designing a task-specific CNN architecture, the option that we have chosen to solve this issue and proposed the LeNet+ architecture.

In the next section, we present our set of experiments with CNN-based cross-modality learning for the pedestrian recognition task. We describe the experimental setup and assess the performance of our approaches.

1.4 Evaluation of Classification Components

In this section, we present our set of experiments including setups and assess the performance of our approaches.

1.4.1 Experimental Setups and Evaluation Protocol

The experiments were analyzed using various dataset and different dataset settings:

1. We used the Caltech dataset using RGB image modality where pedestrian bounding boxes(BBs) are more than 50 pixels. All BB were resized to quadratic size (64 x 64 pixels) to obtain a better performance. This is a preliminary classification experiment in order to make an analogy between AlexNet and LeNet.

2. We used the Daimler stereo dataset with the image scalse of 36 x 84 pixels with a 6-pixel border around the pedestrian image crops as in [EESG10] to allow fair comparisons. Since the results were better than [EESG10] by less than the best Daimler classifier [EG11], we decided to use the original image scales in our next experiments.

3. We used the original Daimler stereo vision dataset with the images scalse of 48 x 96 pixels with a 12-pixel border around the pedestrian images as in [EG11] to allow fair comparisons with this approach. Hence in the following, we detailed the experiments setups using these dataset settings and where it requires, we described the dataset and setting used for each experiment.

The experiments were performed on the original Daimler stereo vision images of 48 x 96 px with a 12-pixel border around the pedestrian images acquired from three modalities: Intensity, Depth, and Optical Flow.

The learning process (training and validation) was performed on 84577 samples (52112 samples of non-occluded pedestrians and 32465 samples of non-pedestrians), based mainly on the holdout validation method involving a single run. The holdout validation method consists in using a part of the learning set as a validation set (75% samples for training, and the remaining 25% samples for validation) to fit the CNN's hyperparameters. The holdout validation is applied in two steps. In the first step, all the hyperparameters: the learning function, the learning rate policy, the initial learning rate and the number of epochs/iterations are optimized for each image modality among Intensity, Depth and Optical Flow. In the second step, several hyperparameters (the learning function and the learning rate policy) are fixed to their optimum values obtained in the first step, while only the most critical ones: the initial learning rate and the number of epochs/iterations are optimized/validated.

We also used the 10-fold cross-validation (CV) to fine-tune these most critical hyper-parameters. The k-fold CV consists of randomly partitioning the training dataset into k=10 equal sized subsamples, then a single one is used to validate

the model, and the remaining subsamples are used for its training. Since this CV method is time costly, only the most critical hyper-parameters (the initial learning rates and the number of epochs/iterations) of the most promising multi-modal In-CML classifiers are optimized.

The testing dataset used to assess the classification performance is independent of the training/validation datasets, and it has the same samples as suggested in the Daimler datasets. It contains 36768 samples of pedestrians (25608 samples of non-occluded pedestrians and 11160 samples of partially occluded pedestrians), and 16235 samples of non-pedestrians.

The learning (training and validation) process for all the proposed CNN models was done with the Caffe Deep Neural Network Framework [JSD+14]. The performances are assessed by the Accuracy (ACC) and the Receiver Operating Characteristics (ROC) curves. The classification task is evaluated also by the area under the curve (AUC). These performance measures are completed with the F-measure to provide the harmonic average of the precision and recall, which is essential for the object detection system design. The ACC, AUC, F-measure values and ROC curves were executed with the Scikit-Learn tool [PVG+11]. We calculate the margin of error (Confidence Interval - CI) with a confidence level of 95% to evaluate whether one model is statistically better than another one. If the CI of two classifiers are disjoint then the one that is significantly statistically better then other can be chosen.

$$\mathrm{CI} = 1.96\sqrt{\frac{\mathrm{P}(100-\mathrm{P})}{\mathrm{N}}}\%. \tag{1.1}$$

In this formulation, P represents the performance of the classification system (e.g., ACC, AUC) computed from the confusion matrix, and N represents the number of testing samples. We also measured the Structural Similarity [WBSS04] and computed the Correlation Coefficient between two couples of images (Intensity-Flow, Intensity-Depth, Depth-Flow) to better analyze the classifier performances and to estimate the area of interest for the proposed cross-modality learning methods.

1.4.2 Evaluation of Uni-Modal Learning Classifiers

In order to test the CNN's performance, we carried out several experiments. In our first experiment, we investigated the performances of AlexNet [KSH12a] and LeNet [LBBH98] on the Caltech [DWSP09] dataset using RGB image modality where pedestrian bounding boxes (BBs) are more than 50 pixels. All BB were resized to quadratic size (64 x 64 px). We observed that the LeNet provides the best results for these small size image datasets (see Table 1.1).

In the second experiment, we evaluated the LeNet architecture with various learning algorithms: Stochastic Gradient Descent (SGD) [Bot12], Adaptive Gradient, RMSPROP [TG12], ADADELTA and learning rate policies: Fixed (FIX), Exponential (EXP) [Sun13], Step Down, Polynomial Decay (POLY) [BT10], Sigmoid, Multi-Step and Inverse Decay. Each modality classifier was exclusively trained with images of its own modality. We used a fixed batch size of 64 images which means that the training set (63433 samples) needs 992 iterations for one epoch. The holdout validation provides the optimal hyper-parameters for the Intensity modality: 29760 iterations and 0.01 initial learning rate using the RMSPROP learning algorithm (RMS-decay, τ=0.98) and POLY (power, ρ=0.75) learning rate policy; for the Depth modality: 29760 iterations and 0.01 initial learning rate using SGD learning algorithm

(gamma, γ =0.99; momentum, μ=0.89) and EXP learning rate policy; for the Optical Flow modality: 29760 iterations and 0.01 initial learning rate using the ADADELTA learning algorithm (momentum, μ=0.89) and FIX learning rate policy. We conclude that various modalities require different learning algorithms and rate policies for efficient learning but an equivalent number of iterations and similar initial learning rates (see Table 1.2. We obtained ACC=96.55% on the Intensity modality (see Table 1.3) followed by the Depth modality with ACC = 89.78% and finally the ACC = 87.34% for Optical Flow.

Table 1.1 – Comparison of learning algorithms and rate policies between AlexNet and LeNet on Caltech dataset

AUC%	STEP		INV		POLY	
SGD	LeNet	95.69%	LeNet	96.24%	LeNet	95.6%
ADAGRAD	LeNet	94.17%	LeNet	92.66%	LeNet	94.64%
ADAM	AlexNet	92.24%	AlexNet	96.%	AlexNet	92.14%
RMSPROP	AlexNet	96.05%	AlexNet	93.05%	LeNet	**97.09%**
NAG	LeNet	96.98%	LeNet	95.82%	LeNet	95.51%

Table 1.2 – Comparison of learning algorithms and rate policies on Intensity, Depth and Optical Flow Daimler data sets

		Accuracy						
Modality Type	Learning rate polics / Algorithm Learning	EXP	FIX	INV	POLY	SIG	STEP	MS
Intensity	SGD	95.96%	96.07%	96.01%	96.09%	96.01%	96.20%	95.78%
	RMSPROP	95.53%	61.19%	95.24%	**96.55%**	96.42%	95.91%	93.37%
	ADADELTA	88.67%	93.08%	91.77%	88.79%	91.96%	91.10%	89.75%
	ADAGRAD	95.02%	95.41%	95.83%	95.49%	95.46%	95.87%	95.02%
Depth	SGD	**89.78%**	61.2%	89.26%	89.69%	88.24%	88.97%	61.2%
	RMSPROP	88.64%	61.17%	81.99%	89.10%	88.66%	89.22%	83.54%
	ADADELTA	87.14%	88.11%	87.64%	87.27%	88.24%	87.72%	87.77%
	ADAGRAD	88.77%	88.81%	89.44%	89.25%	89.44%	89.09%	88.71%
Flow	SGD	86.53%	61.2%	86.69%	86.90%	86.72%	86.84%	61.2%
	RMSPROP	86.89%	61.91%	80.33%	85.69%	87.16%	86.33%	86.57%
	ADADELTA	86.56%	**87.34%**	87.08%	86.78%	87.03%	86.82%	87.18%
	ADAGRAD	87.22%	86.46%	87.11%	86.17%	86.59%	86.68%	86.97%

1.4.3 Evaluation of the Particular Cross-Modality Learning Classifier

We tested the particular cross-modality learning (PaCML) models where each CNN based classifier is learnt on one modality with the holdout validation method, but tested on a different one (see Table 1.3). The best performance for this approach is achieved on Intensity images when trained on Flow images (ACC = 73.79%), on Depth images when trained on Intensity images (ACC = 58.24%), and on Flow images when trained on Intensity images (ACC = 72.97%). The performances below are those obtained when the learning and testing are performed on the same modality. This idea could be promising for the automatic annotation of modality images with a classifier trained with other modality data.

In order to estimate the generalization skills of the proposed automatic annotation approach, we need to know whether this ability depends on the similarity

Table 1.3 – Comparison of Classical Uni-Modal Learning (UML) vs. Particular Cross-Modality Learning (PaCML) on Non-Occluded Pedestrian Daimler Date Set through LeNet CNN architecture with RMSPROP and POLY Learning Settings

	Train on	Valid on	Test on	ACC ± CI
UML	Intensity	Intensity	Intensity	96.550(174) %
	Depth	Depth	Depth	89.100(298) %
	Flow	Flow	Flow	85.690(335) %
PaCML	Depth	Depth	Intensity	50.510(479) %
	Flow	Flow	Intensity	73.790(421) %
	Intensity	Intensity	Depth	58.240(472) %
	Flow	Flow	Depth	54.230(477) %
	Intensity	Intensity	Flow	72.970(425) %
	Depth	Depth	Flow	57.550(473) %

and/or correlation between two modalities. Therefore we compute the Mean of the Structural Similarity Index [WBSS04] (MSSI) (see Table 1.4) and the Mean of the Correlation Coefficient (MR) (see Table 1.5) on the original images and on the edge detector images (using the Laplacian of the Gaussian method) (see Table 1.6) between a pair of images among Intensity-Flow, Intensity-Depth, and Depth-Flow on the training and testing sets.

As reported, (see Tables 1.4, 1.5, 1.6) the Depth with Optical Flow is the most correlated modality pair for MSSI similarity, MR and MR-Log correlation coefficients for all investigated data sets. However, even the highest MSSI similarity between Depth and Optical Flow in the original images is of 0.3319 which proves a low correlation. This highlights the generalization capability of the proposed PaCML model on Daimler [EESG10] dataset.

The MSSI and MR index returns a value in the range from -1 and 1. For MSSI the value of 1 indicates two identical sets of data, and 0 signifies, there is no structural similarity of data. For MR the 1 shows the most robust possible agreement and 0 the most reliable possible.

Nonetheless, the best performance was obtained with the following particular cross-modality models: learnt on Intensity and tested on Flow and respectively learnt on Flow and tested on Intensity. This method raises the question of whether we can regenerate data in one domain by the observation from the other domain. The Depth modality could not be regenerated only from the Intensity modality because two stereo images are needed (space redundancy). The Flow modality could be created from intensity modality if one has access to images from previous times (temporal redundancy).

1.4.4 Comparison of Uni-Modal Classifiers with Cross-Modality Learning Models

In this section, all the models were trained on the LeNet architecture with the same learning algorithm settings (RMSPROP -RMS-decay, τ=0.98 are the optimal ones found previously for the most efficient Intensity modality) and learning rate policy (POLY – power, ρ=0.75), and tested on the non-occluded pedestrian Daimler dataset. The CNNs were enhanced with holdout validation method on the learning set through an optimal number of iterations (29760), and an optimal initial learning

Table 1.4 – Mean of The Structural Similarity Index (MSSI) on the original image Daimler dataset

MSSI	Intensity Depth	Intensity Flow	Depth Flow
Pedestrians Train Sets	0.1430	0.1592	0.3319
Non Pedestrian Train Sets	0.1150	0.1399	0.3213
Non-Occluded Pedestrians Test Sets	0.1335	0.1529	0.3058
Non Pedestrian Test Sets	0.1129	0.1446	0.2865

Table 1.5 – Mean of Correlation Coefficient (MR) on the original image Daimler dataset

|MR|	Intensity Depth	Intensity Flow	Depth Flow
Pedestrian Train Sets	0.0011	0.0117	0.0433
Non Pedestrian Train Sets	0.0575	0.0358	0.1222
Non-Occluded Pedestrians Test Sets	0.0359	0.0077	0.0402
Non Pedestrian Test Sets	0.0170	0.0252	0.0752

rate (0.01) for the classical uni-modal learning method and all cross-modality learning models except for the correlated cross-modality one. Since the complexity of the CNN's learning algorithm for the correlated cross-modality learning was extended, the holdout validation provided an optimal number of training iterations increased to 89220 for an initial learning rate (0.01).

Separate Cross-Modality Learning Approach

We evaluated the separate cross-modality learning models where each CNN-based classifier is trained and tested on one image modality but validated (holdout validation method) on a different one. These experiments prove that the cross-modality learning approach performs slightly better than the classical learning approach (see Table 1.7), but only for the Optical Flow and Depth modalities. The improvements are statistically significant only for Optical Flow ΔACC=0.25% (when validated on Depth). This could be explained by the fact that for the Depth-Flow modality pair, the values of the MSSI, MR, MR-LOG (see Tables 1.4, 1.5, 1.6) are stronger than for the other modality pairs (Intensity-Depth, and Intensity-Flow).

Correlated Cross-Modality Learning

Since the RMSPROP with POLY learning rate settings produced successful results on the Intensity modality, we used those learning settings for all correlated cross-modality (CoCML) models.

The CoCML models are validated following two different approaches: on the multi-modal union dataset or on a uni-modal dataset (see Table 1.7). The multi-modality union validation approach yields better results than the uni-modal validation approach. This method performs better than classical uni-modal learning, but only on the Optical Flow testing set, the improvement is statistically significant with ΔACC=1.927%. The experiment could explain this problem, with vast (three times

Table 1.6 – Mean of the Correlation Coefficient (MR-LOG) on the edge detector images Daimler dataset, using the Laplacian of Gaussian method

MR-LOG	Intensity Depth	Intensity Flow	Depth Flow
Pedestrians Training Sets	0.0126	0.0106	0.0178
Non Pedestrians Training Sets	0.0128	0.0111	0.0253
Non-Occluded Pedestrians Test Sets	0.0142	0.0085	0.0149
Non Pedestrians Test Sets	0.0154	0.0139	0.0198

more) and different modalities (Intensity, Depth, and Optical Flow) training data, the breadth and depth of the network should be extended. Moreover, according to [LBBH98], the complexity would be limited by the computing resources, which would thus hinder the performance.

Table 1.7 – Comparison of Correlated (CoCML), Separate (SeCML) vs Incremental Cross-Modality (InCML) Learning Models on the Non-Occluded Daimler Pedestrian Dataset. The results in bold are statistically better than those obtained with the Classical Uni-Modal method.

CNN	Learning Settings	Validation Method	Approach	Train on	Valid on	Test on	ACC ± CI
LeNet	Same Settings for RMSPROP with POLY	Holdout	Classical Uni-modal	Intensity	Intensity	Intensity	96.550(174) %
				Depth	Depth	Depth	89.100(298) %
				Flow	Flow	Flow	85.690(335) %
			SeCML	Intensity	Depth	Intensity	96.310(180) %
				Intensity	Flow	Intensity	96.230(182) %
				Depth	Intensity	Depth	89.000(299) %
				Depth	Flow	Depth	89.330(338) %
				Flow	Intensity	Flow	86.120(331) %
				Flow	Depth	Flow	**86.600% ± 0.325%**
			CoCML	Intensity_i+Depth_i+Flow_i i=$\overline{1,n}$	Intensity_j+Depth_j+Flow_j j=$\overline{1,m}$	Intensity	94.540(217) %
					Intensity_j+Depth_j+Flow_j j=$\overline{1,m}$	Depth	85.390(338) %
					Intensity_j+Depth_j+Flow_j j=$\overline{1,m}$	Flow	**88.26% ± 0.308%**
					Intensity	Intensity	94.400(220) %
					Depth	Depth	86.060(331) %
					Flow	Flow	**87.38% ± 0.318%**
			InCML	Depth_i,Flow_i,Intensity_i i=$\overline{1,n}$	Depth_j,Flow_j,Intensity_j j=$\overline{1,m}$	Intensity	96.700(171) %
				Intensity_i,Flow_i,Depth_i i=$\overline{1,n}$	Intensity_j,Flow_j,Depth_j j=$\overline{1,m}$	Depth	89.390(295) %
				Intensity_i+Depth_i+Flow_i i=$\overline{1,n}$	Intensity_j+Depth_j+Flow_j j=$\overline{1,m}$	Flow	**87.02% ± 0.3220%**
	Optimal Specific Settings	K-fold Cross Validation K=10	InCML	Depth_i,Flow_i,Intensity_i i=$\overline{1,n}$	Depth_j,Flow_j,Intensity_j j=$\overline{1,m}$	Intensity	**97.50%± 0.149%**
				Intensity_i,Flow_i,Depth_i i=$\overline{1,n}$	Intensity_j,Flow_j,Depth_j j=$\overline{1,m}$	Depth	88.92% ± 0.300%
				Intensity_i+Depth_i+Flow_i i=$\overline{1,n}$	Intensity_j+Depth_j+Flow_j j=$\overline{1,m}$	Flow	**88.70% ± 0.303%**
LeNet+	Optimal Specific Settings	K-fold Cross Validation K=10	InCML	Depth_i,Flow_i,Intensity_i i=$\overline{1,n}$	Depth_j,Flow_j,Intensity_j j=$\overline{1,m}$	Intensity	**97.78% ± 0.141%**
				Intensity_i,Flow_i,Depth_i i=$\overline{1,n}$	Intensity_j,Flow_j,Depth_j j=$\overline{1,m}$	Depth	**91.30% ± 0.27%**
				Intensity_i+Depth_i+Flow_i i=$\overline{1,n}$	Intensity_j+Depth_j+Flow_j j=$\overline{1,m}$	Flow	**89.75% ± 0.29%**

Incremental Cross-Modality Learning

Since the incremental cross-modality learning (InCML) method is the most promising approach, we decided to carry out more extensive experiments. Thus, the InCML models were learnt using different approaches:

Table 1.8 – Optimal Learning Rate and number of iterations for the Incremental Cross Modality Learning with K=10 cross-validation for LeNet and LeNet+ Architectures on Daimler dataset

Image modality	CNN	Initial Learning Rate		Iterations
		Specific	Averaged	
Intensity	LeNet	0.01	1.5e-05	158640
	LeNet+	0.001	1.2e-05	119040
Depth	LeNet	0.01	1.93e-04	208320
	LeNet+	0.001	1.014e-05	208320
Optical Flow	LeNet	0.01	1.5e-04	158640
	LeNet+	0.01	1.2e-05	158640

(a) Training and holdout validation using the same settings for the learning algorithm (RMSprop with RMS-decay, τ=0.98), for the learning rate policy (POLY with power, ρ=0.75) and a batch size=64 for all three modality-specific CNNs;

(b) Training and holdout validation using optimal modality-specific hyper parameter settings for each CNN. For the Intensity modality: RMSPROP with RMS-decay, τ=0.98 and POLY with power, ρ=0.75; for the Depth modality: SGD with gamma, γ=0.99; momentum, μ=0.89 and EXP; for the Optical Flow modality: ADADELTA with momentum, μ=0.89 and FIX learning rate policy (see Section V.B);

(c) Training and k-fold cross-validation method using the algorithm settings from point (a);

(d) Training and k-fold cross-validation method using the algorithm settings from point (b);

The holdout validation in (a) and (b) approaches makes it possible not only to fit the optimal initial learning rate, but also to verify that 29760 iterations avoid under and over fitting. The k-fold cross-validation in (c) and (d) approaches started learning with specific initial learning rates for each modality CNN based on LeNet and respectively LeNet+ architecture for all ten train/valid folds. For each fold and modality CNN we considered the final learning rate for 29760 iterations. The optimal initial learning rate value for each modality CNN are obtained by averaging the final values from prior training folds. These optimal values are used to initialize the training of each modality CNN in a holdout validation method. This makes it possible to find out the optimal number of iterations for the last CNN within each InCML model. The optimal hyperparameter values used in the last learning process are depicted in Table 1.8.

As shown in Table 1.7, the InCML learning approach based on the LeNet architecture with the holdout validation method and RMSPROP - POLY settings, performs slightly better than classical uni-modal learning for all image modalities, but the improvements are statistically significant only for the Optical Flow modality. The LeNet+ architecture we have proposed, with the K-fold cross-validation method and optimal specific learning settings, performs better than the classical learning approach, for all image modalities and the improvements are statistically significant for all image modalities: ΔACC_I =0.915%, ΔACC_D=1.632%, ΔACC_F=3.435% (see Table 1.7). Moreover, this approach is more flexible, allowing for adaptive settings

according to each CNN classifier whereas the correlated cross-modality method requires using a single CNN model and therefore the same learning settings.

1.4.5 Early-Fusion vs Late-Fusion with Classical Learning method

The training and testing were carried out on Daimler stereo vision images of 36 x 84 pixels with a 6-pixel border around the pedestrian image crops in three modalities: Intensity, Depth and Optical flow, to allow fair comparisons [EESG10].

We use 84577 samples for training, 75% of which were for learning, 25% for validation and 41834 for testing. The best performances optimized on the validation set were acquired with 208320 epochs and 0.01 learning rate.

	Intensity	Depth	Flow	Early-Fusion	Late-Fusion
AUC	96.39%	87.08%	88.24%	89.88%	99.21%
IC/2	±0.08	±0.15	±0.16	±0.14	±0.04

Table 1.9 – Single-modality vs. multi-modality on Daimler testing set using images of 36 x 84 pixels.

In Table 1.9 we show the AUC obtained with single-modality versus multi modality. The best performance with single modality is obtained for intensity (96.39%) followed by Depth and Optical Flow. For the multi-modality architectures, the late-fusion solution we propose not only outperforms all the single modality classifiers but also the early fusion solution. This improvement is statistically significantly since the confidence intervals are disjoint. These performance are also shown in the ROC curves (see Fig 1.11).

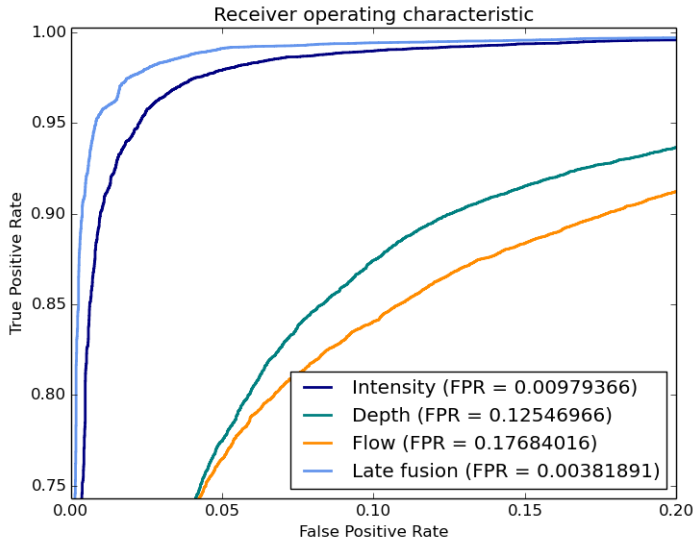


Figure 1.11 – Single-modality vs. multi-modality ROC classification performance on Daimler testing dataset using images of 36 x 84 pixels. FPR at 90% detection rate.

The classifier system proposed in [EESG10] obtains an FPR of 0.0125% at 90% detection rate with their SVM early-fusion model on Daimler dataset. Therefore our late-fusion model significantly improves the classification performance with Δ FPR=0.0085%. However the best classifier on the Daimler dataset carried out with

HOG+LBP/MLP, Monolithic HOG classifier (Intensity + Depth + Flow) [EG11], allowed an FPR of 0.00026% at 90% detection rate. This system used features, which are extracted explicitly and made the training more efficient while the CNN classifier has to extract implicit features and requires several samples for training.

Since the SVM late fusion model did not exceed the best Daimler pedestrian classifier, we picked to handle only the MLP as in [EG11] in our next late-fusion analyses to enable for an evenhanded comparison.

1.4.6 Late-Fusion with Classical vs Cross-Modality Learning using LeNet and LeNet+ CNN architectures

The late fusion approach was investigated in two ways using MLP: with the LeNet and respectively with the LeNet+ architecture we proposed, both on the non-occluded pedestrian and also on the partially occluded pedestrian data sets.

First, all the models were trained using the same settings with the incremental cross-modality approach. We chose the RMSPROP with the POLY settings since they allowed for the best results on the Intensity modality. Then, optimal specific parameters (selected from the validation set) were used to learn the CNNs through the incremental cross-modality model, and consequently the RMSPROP with the POLY settings for the Intensity modality, the SGD with EXP settings for the Depth modality and ADADELTA with FIX settings for the Flow modality.

The LeNet+ architecture performs statistically better than LeNet for both non-occluded (see Table 1.10) and partial-occluded (see Table 1.11) Daimler dataset not only with classical learning but also for the incremental cross-modality learning. Indeed the confidence intervals are disjoint.

In Tables 1.10 and 1.11 we show that the performance obtained with incremental cross-modality using the best specific modality learning settings are statistically better than those obtained with the same learning settings. The incremental cross-modality learning is an efficient solution not only with the single modality classifiers, but also with the late-fusion scheme, since its performance is statistically better than late fusion with classical learning.

The improvements brought by non-occluded pedestrian and partially occluded pedestrian over the Daimler datasets are respectively:

- $\Delta\text{ACC}_{non-occluded} = 1.1034\%$;
- $\Delta\text{AUC}_{non-occluded} = 1.5047\%$;
- $\Delta\text{ACC}_{partially-occluded} = 5.281\%$;
- $\Delta\text{AUC}_{partially-occluded} = 5.497\%$

The improvements are higher for partially-occluded pedestrian recognition than for non-occluded pedestrian recognition. This result proves the robustness of our models. These assessments are also drawn in the ROC curves (see Figure 1.12). We observe that the ROC curves obtained with the InCML based models with K-Cross and specific settings (SP) are statistically better than all the other approaches but the improvement obtained with LeNet+ vs LeNet is limited.

Table 1.10 – The performance with late fusion on Non-Occluded Pedestrian Daimler Testing Set. The results in bold are statistically better than those obtained with Classical Uni-Modal Learning. SM=Same Settings, SP=Specific Settings, K-Cross=K-fold Cross-Validation.

CNN	Late-fusion	Trained on	AUC ± CI	ACC ± CI	F1-Measure ± CI
LeNet	Classical Learning	SM	97.040(162) %	97.460(150) %	97.3100(1609) %
LeNet+		SM	97.560(153) %	97.970(140) %	97.4600(1565) %
LeNet	Incremental Cross Modality Learning	SM	97.200(158) %	97.620(146) %	97.4900(1556) %
		SP	97.47(15) %	97.690(143) %	97.5400(1540) %
		SP; K-Cross	*98.26% ± 0.125%*	*98.29%± 0.124%*	*98.60% ± 0.1168%*
LeNet+	Incremental Cross Modality learning	SP; K-Cross	***98.811% ± 0.1039%***	***98.817% ± 0.1036%***	**99.11% ± 0.0934%**

Table 1.11 – The performance with late fusion on Partially Occluded Pedestrian Daimler Testing set. The results in bold are statistically better than those obtained with Classical Uni-Modal Learning. SM=Same Settings, SP=Specific Settings, K-Cross=K-fold Cross-Validation.

CNN	Late-fusion	Trained on	AUC ± CI	ACC ± CI	F1-Measure ± CI
LeNet	Classical Learning	SM	78.130(489) %	81.110(463) %	80.6600(4677) %
LeNet+		SM	84.930(423) %	82.490(450) %	82.4800(4502) %
LeNet	Incremental Cross Modality Learning	SM	78.360(487) %	80.480(469) %	79.5700(4775) %
		SP	78.400(535) %	81.300(461) %	80.8700(4658) %
		SP; K-Cross	*82.88% ± 0.446%*	*85.09% ± 0.421%*	*84.65% ± 0.4269%*
LeNet+	Incremental Cross Modality Learning	SP; K-Cross	***86.12% ± 0.409%***	***88.38% ± 0.379%***	**88.34% ± 0.3801%**

1.4.7 Late-Fusion with Classical vs Incremental Cross-Modality Learning using AlexNet and VGG-16 CNN architectures

The architecture involves three independent InCML-based classifiers which are fed with a specific modality among Intensity, Depth and Optical Flow, and an MLP which discriminates between pedestrians (P) and non-pedestrians ($\overline{P}$) using probabilistic class estimates provided by each InCML classifier. The last layer of each CNN provides the probability output scores of Intensity Pr(I), Depth Pr(D) and Optical Flow Pr(F). The MLP includes three neurons in the input layer, one hidden layer of 100 neurons, and 2 neurons in the output layer. The ReLU function with a Stochastic Gradient Descent solver and a constant learning rate of 1e-07 were used for the weights optimization.

The learning and testing were carried out on Daimler [EESG10] stereo vision images of 48 x 96 pixels with a 12-pixel border around the pedestrian images extracted from three modalities: Intensity, Depth and Optical Flow.

The learning and testing processes operate on AlexNet and VGG-16 using original (default) and optimized settings respectively. For the learning with the incremental cross-modality and classical methods, we use RMSPROP with POLY settings for all CNNs with the same learning rate (0.0001). The setting optimization consists in removing the crop of size features, reducing the number of outputs (from 96 to 20), decreasing the kernel size (from 7 to 5) and minimizing the stride (from 4 to 1 for AlexNet and from 2 to 1 for VGG-16) in the first convolution layer. We also marked down the number of outputs (from 4096 to 2048) in the last two Fully Connected (FC) layers for AlexNet and respectively in the previous three FC layers (from 4096 to 2048 in the FC6 and FC7 respectively from 1000 to 500 in the FC8) for VGG-16.

The complexity of the classification system is assessed by the False Positive Rates (FPR) using a True Positive Rate (TPR) of 95% and a Confidence Interval (CI) to prove whether one model is statistically better than another one. The results, given in

Table 1.12 – Comparison of our models with the state-of-the-art with the false positive rate at 95% True Positive Rate on Daimler dataset

Method	Pedestrian DataSet	FP Rate ± CI
Deep DP-BM [OW12]	Partially Occluded	0.25 ± 0.0043%
HOG/linSVM MoE [EESG10]		0.20 ± 0.0040%
L+; InCML; K-Cross; SP		**0.124 ± 0.0033%**
Deep DP-BM [OW12]	Non Occluded	0.05 ± 0.0021%
HOG/linSVM MoE [EESG10]		0.0302 ± 0.0016%
HOG+LBP/MLP MoE [EG11]		0.0035 ± 0.00056%
L+; InCML; K-Cross; SP		**0.0016 ± 0.000382%**

Table 1.13, allow for the following comparisons on the Daimler data sets:

- Default vs. optimized settings: The optimization method presented allows statistically significant improvement for both CL/InCML methods and AlexNet, VGG-16 architectures up to Δ= 0.4925%

- Incremental Cross-Modality Deep Learning (InCML) vs. Classical Learning (CL): With the optimized settings the results obtained with InCML are statistically better than those achieved with the CL, but only with AlexNet on the partially occluded pedestrian Daimler dataset and with VGG-16 on the non-occluded pedestrian Daimler dataset.

1.4.8 Comparisons with the state-of-the-art methods

We choose to compare our best classifier LeNet+ with Incremental Cross-Modality learning with specific learning settings and the K-fold Cross Validation method (L+; InCML; K-Cross; SP) with the state-of-the-art classifiers provided on the Daimler data sets. These classifiers are based on a mixture of experts (MoE) with handcrafted features HOG/linSVM [EESG10] and respectively HOG+LBP/MLP [EG11] within a late fusion of Intensity, Depth and Optical Flow modalities. We also considered for comparison the best Deep models provided on the Daimler dataset, based on Deformation Part and Boltzmann Machine (Deep DP-BM) [OW12].

For the comparison, we cannot draw the ROC curves of these classifiers since the algorithm's source code is not provided, nor is a detailed explanation of the learning methodology given. Thus, no information is given concerning the learning settings for MLP (e.g., learning rate, number of iterations), nor for SVM (e.g., penalty parameter C of the error term, tolerance for stopping criteria, loss function) or how those hyper-parameters were optimized. Since we do not know how the learning set was shared between the training and validation sets and whether a cross-validation or a holdout validation technique was used, we cannot reproduce the classification method in a fair manner.

Therefore, to assess the performance of our best classifier (L+; InCML; K-Cross; SP), we compute the false positive rates (see Table 1.12) using a true positive rate of 95% as a frequent reference point using the interpolation method. This target allows a fair comparison with the cited state-of-the-art pedestrian classifiers on both

Table 1.13 – Comparison of AlexNet and VGG-16 with the state-of-the-art on Daimler dataset

Pedestrian Dataset	Method and Settings		TPR 90%	TPR=95% FPR ± CI
Partially Occluded (p-occ)	AlexNet	Default-CL	0.73	0.8671 ± 0.0034%
		Default-InCLM	0.712	0.7126 ± 0.0037%
		Optim-CL	0.137	0.2363 ± 0.0042%
		Optim-InCLM	0.105	**0.1920 ± 0.0039%**
	VGG-16	Default-CL	0.597	0.7360 ± 0.0044%
		Default-InCLM	0.605	0.7704 ± 0.0042%
		Optim-CL	0.447	0.6495 ± 0.0047%
		Optim-InCLM	0.457	0.6714 ± 0.0047%
	HOG/linSVM MoE [EESG10]		0.175	0.20 ± 0.0040%
	Deep DP-BM [OW12]		0.216	0.25 ± 0.0043%
Non Occluded (non-occ)	AlexNet	Default-CL	0.328	0.4465 ± 0.0048%
		Default-InCLM	0.362	0.4939 ± 0.0048%
		Optim-CL	0.0006	**0.0011 ± 0.00031%**
		Optim-InCLM	0.0009	0.0014 ± 0.00035%
	VGG-16	Default-CL	0.151	0.2531± 0.0042%
		Default-InCLM	0.125	0.2150± 0.0039%
		Optim-CL	0.011	0.0296± 0.0016%
		Optim-InCLM	0.0078	0.0236± 0.0015%
	HOG+LBP/MLP MoE [EG11]		0.0002	0.0035 ± 0.00056%
	HOG/linSVM MoE [EESG10]		0.011	0.0302 ± 0.0016%
	Deep DP-BM [OW12]		0.007	0.05 ± 0.0021%

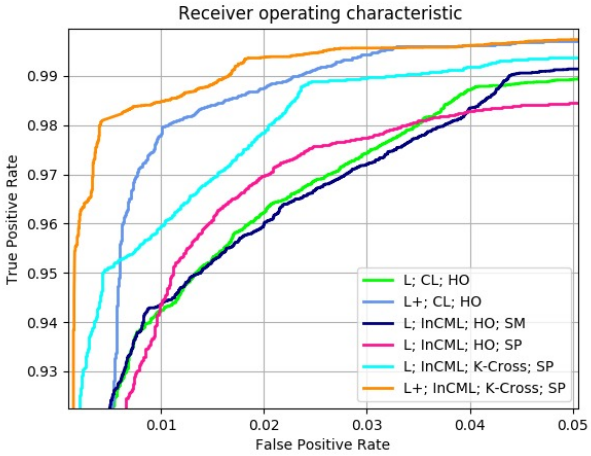
partially-occluded and non-occluded pedestrian Daimler data sets. We also computed the confidence intervals (CI) with a risk level of 0.05 to allow a significant statistical analysis. Our model outperforms both the handcrafted-features MoE and deep DP-BM models.

The improvements obtained with our classifier (L+; InCML; k-Cross; SP) compared with all these models are statistically significant on both partially occluded and non occluded data sets since the confidence intervals are disjoint:

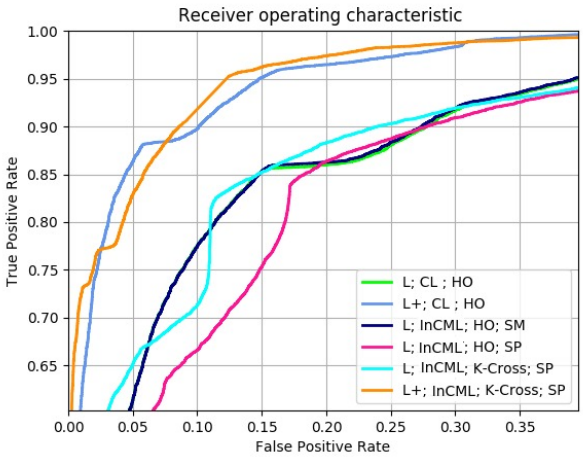
- ΔFPR MoE$_{partially-occluded}$=0.0484%;
- ΔFPR MoE$_{non-occluded}$=0.0019%;
- ΔFPR DP-BM$_{partially-occluded}$=0.126%;
- ΔFPR DP-BM$_{non-occluded}$=0.076%;

It is interesting to note that the improvement obtained with our model is more significant for the partial-occluded task for both the handcrafted-features and deep models. However, our model needs to be validated on more extensive datasets and various applications (multiclass road obstacle detection, traffic collision risk assessment).

The improvements obtained with optimized settings on TPR=95% based on AlexNet using the InCML method approach are statistically significant on both partially-occluded and non-occluded data sets since the confidence intervals are disjoint:



(a) Non Occluded Pedestrian Data Classification



(b) Partially Occluded Pedestrian Data Classification

Figure 1.12 – The ROC classification performance on Daimler testing dataset; where L=LeNet Architecture, L+= Extended LeNet Architecture, CL=Classical Learning method, HO=Holdout validation, SM=Same Settings, SP=Specific Settings, K-Cross=K-fold Cross-Validation.

- ΔFPR MoE$_{p-occ}$=0.008%;
- ΔFPR MoE$_{non-occ}$= 0.0024%;
- Δ FPR DP-BM$_{p-occ}$=0.0489%;
- ΔFPR DP-BM$_{non-occ}$ =0.076%.

On the other hand, the method used AlexNet and VGG-16 do not outperform the method [EESG10] at TPR=90%. However, [EESG10] was only analyzed on the non-occluded pedestrian dataset. Our approach was learnt on the entire dataset which includes occluded and non-occluded pedestrian samples. It is to be noted that the AlexNet obtained better results than VGG-16 on the Daimler data sets, this highlighting that a huge architecture does not always achieved better results.

1.5 Conclusion

In this chapter, we systematically depicted different cross-modality learning approaches of various methods based on Convolutional Neural Networks for pedestrian recognition:

- We studied three methods of integrating different image modalities (Intensity, Depth, Optical Flow) to improve the classification component. Two methods of training were analyzed:
  1. Classical Training;
  2. Cross Modality Training;
- We studied how learning representations from one modality would enable prediction for other modalities, which is termed cross-modality learning. Four approaches were proposed:
  1. a particular cross-modality learning (PaCML);
  2. a separate cross-modality learning (SeCML);
  3. a correlated cross-modality learning (CoCML);
  4. an incremental cross-modality learning (InCML);
- We studied two different fusion schemes:
  1. Early Fusion;
  2. Late Fusion;

We presented an early fusion versus late fusion comparison on the non-occluded Daimler stereo vision dataset. The early fusion approach integrates three image modalities (Intensity, Depth and Optical Flow) by concatenating them to learn a single CNN. The late fusion approach consists in fusing the probabilistic output scores of three independent CNNs, trained on different image modalities (Intensity, Depth and Optical Flow) by an MLP classifier.

We concluded that the early-fusion approach is less efficient and robust than the late-fusion model. Moreover, the early-fusion model requires high image calibration and synchronization. The early-fusion training method is more constrainable since, for a given image frame, it needs an item for each modality, and therefore the classifier requires more samples to learn the problem. With the early-fusion model, it is impossible to take advantage of cross-dataset training methods, by using modality images from different unimodal and/or multi-modal datasets where all the modalities involved are not acquired and/or annotated. The early fusion method does not allow one to improve the learning by extending the number and the variety of items through cross-modality learning.

The particular cross-modality learning could be extended for an automatic annotation method of new modality images. Incremental cross-modality learning could be used when there are not enough annotated images in each modality to improve the classification performances. The separate and correlated cross-modalities learning models do not allow for statistically significant improvements since they require the same learning settings for all modality models and, for the second one (CoCML), the same image frame for each modality.

The effectiveness of those methods has been analyzed through various performance measures with statistical coefficients (Confidence Intervals, Correlation Coefficients, Structural Similarity Index). Incremental cross-modality learning based on modality transfer learning is better than both the separate and correlated cross-modality learning models. It also improves the classification performances, in contrast to classical learning of uni-modal CNNs, through late-fusion designed on the Daimler dataset. We assume that the incremental method is the promising cross-modality learning model. Indeed, this cross-modality learning method is more flexible than the others we analyzed since it could be used with different learning settings adapted for each image modality. In order to improve its performances, we proposed a new CNN architecture called LeNet+ which outperforms the state-of-the-art pedestrian classifier for both non-occluded and partially-occluded pedestrian Daimler dataset. However, those cross-modality learning methods have to be validated not only for pedestrian classification, but also for pedestrian unit action recognition, pedestrian detection and tracking.

The enhancements proposed in LeNet+ allow us to validate the cross-validation learning methodology and chose from the proposed models (PaCML, SeCML, CoCML, InCML) the most promising one on a multi-modality classification task on the Daimler dataset. The InCML model could be used not only for an ADAS system but also for a wide variety of learning components with a multi-modality system within complex multi-class classifiers.

Future work, we are planing to work with CNNs designed for multi-class detection (SSD, Faster RCNN, R-FCN) on different databases. In addition, we intend to apply the promising InCML model for the classification and detection of other road objects (traffic signs and traffic lights) and road users (vehicles, cyclists).

In the next Chapter, the Incremental Cross-Modality Deep Learning approach is applied in the pedestrian detection task and pedestrian action recognition issue. Its evolution and advancement using complicated and huge detector CNN has been deeply investigated.

Chapter 2

Pedestrian Detection with Action Classification

Contents

2.1 Introduction **43**

2.2 Related Work **46**

2.2.1 Object Detectors 46

2.3 Pedestrian Detection **51**

2.3.1 Pedestrian Detection Component 51

2.3.2 Depth Modality from JAAD Dataset 52

2.3.3 Optical Flow Modality from JAAD Dataset 53

2.4 Experiments **54**

2.4.1 Data setup 54

2.4.2 Training protocol 55

2.4.3 The convolution neural network setups 56

2.4.4 Testing protocol 57

2.4.5 Evaluation protocol 57

2.5 Evaluation and Results **58**

2.5.1 Evaluation of the Uni-Modal Pedestrian Detection Component 58

2.5.2 Evaluation on Uni-Modal Incremental Cross-Modality Deep Learning Pedestrian Detection 59

2.5.3 Evaluation of Uni Modal Pedestrian Action Detection 61

2.5.4 Evaluation of Incremental Cross Modality Deep Learnig Pedestrian Action Detection 63

2.5.5 Comparison of the Uni-Modal vs Incremental Cross Modality Deep Learning Pedestrian Detection for Pedestrian Action Detection 64

2.6 Conclusion **70**

2.1 Introduction

The ability to detect and classify objects is the fundamental requirement for designing intelligent application systems, like autonomous vehicles, and driver assistance systems.

In the first chapter, we analyzed approximately the first two components from our main objective: the Perception and the Identification/Fusion. The Perception component involves the stereo vision dataset based on the Daimler dataset [EESG10] while the Identification/Fusion component use the environment information provided from the prior component and classify the information, (distinguish between pedestrian and non-pedestrian) based on our cross-modality deep learning approaches. The Identification/Fusion component involves even detection task. Detection is a process that involves the classification task in order to categorize and locate all known content in the scene. Thus, in this chapter, we study the pedestrian detection component, including not only the pedestrian detection task but also the pedestrian action-unit classification using our Incremental Deep Learning Approach, described in the previous chapter (see Figure 2.1).

The detection component raises several problems like viewpoint variation, illumination, occlusion, scale, deformation, background clutter, and intra-class variation. The objects of interest appear in highly dynamic and cluttered environments and have a wide range of looks, due to body size and posture, clothing, viewpoint, and outdoor lighting conditions. The pedestrians must be detected even if they stand far away from the camera, and thus appear rather small in the image, at low resolution. A significant complication comes from the moving vehicle when one does not have the luxury to use simple background subtraction methods (such as those used in surveillance applications) to obtain a foreground region containing the moving obstacle (i.e. pedestrian, vehicle). Furthermore, pedestrians can exhibit highly irregular motion.

Different pedestrian detectors have been developed until now, but in current research, deep learning neural networks, including Convolutional Neural Networks (CNNs), like Fast R-CNN [Gir15], Faster R-CNN [RHGS15a], YOLO [RDGF15], SSD [LAE+15] have frequently led to an enhancement in detection performance, due to their discriminatory features for each raw pixel proposed.

Furthermore, the deep learning object detection mentioned above jointed with differnet backbones (e,g, VGG [SZ14], AlexNet [KSH12a], RestNet [HZRS15], GoogleNet [SLJ+14]), have obtained the best performances for object detection issue.

The difficulty of deep learning approaches is that they require significant data samples and computing power, particularly for the off-line learning process, but also, to a lesser extent, for the on-line detection applications.

A considerable dataset is needed not only to train and test the detection or the classification system components but also to measure its overall performance. The dataset has to cover a broad range of pedestrian appearances captured in diverse environmental conditions: winter, spring, summer, autumn, heavy rain, fog, with different illumination/contrast levels that can appear due to the position of the sun (sunrise, middle of the day, sunset, shadows). Different occlusion situations give special cases that must be captured in the dataset. Hence, we use the JAAD dataset because its pedestrians are annotated from various environmental conditions.

A real-world pedestrian detector has to take into account that most pedestrians (70%) are occluded in at least one frame (especially children in pushchair and old

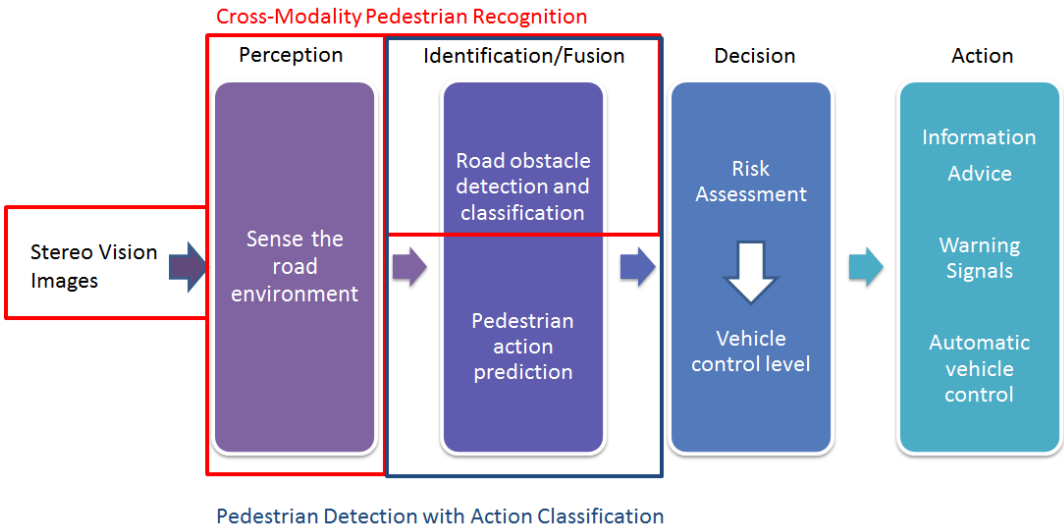


Figure 2.1 – The main architecture of our system. In red are the issues investigated in the first chapter. In blue is the problems studied in this chapter.

persons), underscoring the importance of detecting occluded people; the fraction of occlusion can vary significantly (heavy 35%, full 44%); over 97% of occluded pedestrians belong to just a small subset of the hundreds of possible occlusion types. In this chapter, we use out Incremental Deep Learning (presented in the first chapter) in order to improve the detection/classification performance for occluded and partial occluded based on JAAD [KRT16] dataset.

Since we are planning to use our Incremental Cross modality deep learning approach in the detection task, we need to use three image modality (Intensity, Depth, and Optical Flow). The main problem of the JAAD [KRT16] dataset is that it provides only the RGB image modality. This limitation leads us to use the RGB instead of Intesity and to find a method to extract the Optical Flow and Depth from JAAD dataset. This problem is not very difficult for Optical Flow since we have the frames T and T-1 while for Depth image modality usually needs a stereo camera. Nevertheless, we wondered could we extract the Depth modality using the frames T and T-1 from JAAD dataset? How this representation influences the detection/classification performance using the InCML approach? Hence in this chapter, we investigate our InCML approach using RGB, Optical Flow, and Depth for the detection issue and also validate this method for the detection task.

More than that, in this chapter we propose even a pedestrian detection system able also to recognize pedestrian actions, which is in contrast to classical systems which only discriminate between pedestrians and non-pedestrians from other road users.

This approach is a preliminary experiment before the pedestrian action prediction task, the reason that we investigate it independently. We consider the pedestrian action detection task necessary, even if it detects the action pedestrian at time T=0 because it could offer pertinent information to the diver/car to act in accordance with the pedestrian's action at that time. For instance, the pedestrian sits on a bench, and the car usually moves on the road. In this case, the detector returns the pedestrian's action at the current moment. The pedestrian's action does not represent a dangerous situation so that the car can continue its moves without any limitation. On the other hand, there is also a dangerous situation where the

pedestrian could stand in the middle of the road. To avoid a collision in this dangerous situation, it is necessary to take temporary account information, but for this problem, we investigate in the next chapter.

We wondered what happens if the pedestrian is walking along the street (there is no footpath/crosswalk) or the pedestrian already passed the road, and maybe he/she suddenly decide to return. We wondered if could we make a clear distinction between the pedestrians who are planning to cross the street and then cross the road and the pedestrian who starts to cross the street but for a different reason he/she stops.

We consider that the JAAD annotations do not adequately cover all the pedestrian action/intention. These particularly cases struggle us to derivate our annotation in order to fulfill our thesis objectives. Therefore, according to the specifications and annotations, we separate the pedestrian labels into four classes: pedestrian is preparing to cross the street, pedestrian is crossing the street, pedestrian is about to cross the street, and pedestrian's intention is ambiguous.

The contribution of this Chapter concerns detecting the pedestrian and pedestrian actions. To do so, we develop the following methodology relying on a deep learning approach:

- The RetinaNet [LGG+17a] and Faster R-CNN with Inception V2 [HRS+16a] are trained for pedestrian detection proposes using all pedestrian samples;
- Split the pedestrian Joint Attention for Autonomous Driving (JAAD) [KRT16] dataset into four classes: pedestrian is preparing to cross the street, pedestrian is crossing the street, pedestrian is about to cross the street, and pedestrian's intention is ambiguous.
- Pull out the Optical Flow (T, T-1) and Depth (T, T-1) from JAAD dataset.
- Train all pedestrian samples using the pedestrian action tags mentioned above with the RetinaNet using RGB, Optical Flow and Depth motion for pedestrian detection and action classification;
- Training the Incremental Cross-modality Deep Learning using RetinaNet for pedestrian detection and pedestrian action recognition using RGB, Depth and Optical Flow.

The Chapter is organized as follows: Section 2 outlines some existing approaches from the literature and gives our main contribution. Section 3 presents an overview of our system. Section 4 describes the experiments and the results on the JAAD dataset. Finally, Section 5 presents our conclusions.

2.2 Related Work

Pedestrian detection has been widely investigated in various research tasks because is one of the most significant issues in self-transportation and driver assistance systems [BOHS14, ZBO+16].

These detectors can be analyzed from different points of view since they are situated at the confluence among Machine Learning, Computer Vision and Intelligent Vehicles domains.

The most detector algorithm used in the computer vision is the sliding window. In the context of computer vision, a sliding window is a rectangular region of fixed width and height that slides several windows across an image. For each of these windows, we employ an image classifier to determine if the window has an object that interests us.

Pedestrian detection approaches using sliding window algorithm and various predefined feature vector models (handcrafted feature models) such as Local Binary Patterns (LBP) [BJNL13], Scale Invariant Feature Transform [VGVZ09b], Histograms of Oriented Gradients (HOG) [DT05a], Integral Channel Features [DTPB09] followed by a trainable classifier such as a Support Vector Machine (SVM) [FGMR10b], Multi-Layer Perceptron (MLP), boosted classifiers [DTPB09], were the main methods used for object detection and classification, until progress in deep learning outperformed them for image classification issue [KSH12a].

Deep learning approaches do not require predefined features due to their ability to learn features directly from the images. In deep learning methods, an outstanding representation of the training set is more significant than predefined features to obtain the high-grade performance for the target application.

Due to the possibility of computing parallelizing the learning process on GPU platforms, the interest of using deep learning procedure in pedestrian detection task has increased significantly in the last years.

Therefore, we review the main object detectors based on CNN as Fast R-CNN [Gir15], R-CNN [GDDM13], Faster R-CNN [RHGS15a], YOLO [RDGF15], Single Shot Multibox Detector (SSD) [LAE+15], and Region-Based Fully Convolutional Networks (R-FCN) [DLHS16], followed by some related pedestrian detection work based on deep learning approaches.

2.2.1 Object Detectors

The main difference between object detection and classification algorithms is that detection algorithms try to locate the object of interest by drawing a bounding box around it, in contras with classification algorithms which only name the object of interest. The object detector algorithm takes various regions of interest from each image, and links them up with a CNN classifier, which identifies the object within that region. This approach could be very costly because the algorithm must localize/draw a bounding box for each object found in the image, and it would not know how many objects have already been detected.

To avoid the problem of picking a considerable number of regions, the **R-CNN** [GDDM13] uses a selective search algorithm to extract only 2000 regions (called region proposals) from a given image frame. These regions are covered into a square and inserted into a CNN that renders a 4096-dimensional feature vector as output, and then the outputs are inserted into an SVM to classify the object from the pro-

posed region. However, the R-CNN is limited by the selective search algorithm (it is a fixed algorithm) which is costly in terms of training time and real-time processing.

Fast R-CNN [Gir15] managed to solved a few of the R-CNN's drawbacks and made the R-CNN faster. The method is related to the R-CNN algorithm, where instead of supplying the region proposals to the CNN, it provides the input image to the CNN to create a convolutional feature map. The region proposals are distinguished from convolutional feature map, with a RoI pooling layer. It warps them into squares and insert into a fully connected layer. Then, it uses a softmax layer to predict the class of the proposed region and the offset values for the bounding box. Although the Fast R-CNN is indeed faster than R-CNN, its performance is hindered by the region proposals.

Faster R-CNN [RHGS15a] eliminated the selective search algorithm and used a separate network to predict the region proposals, the region proposal network (RPN). This change makes it faster than previous detectors, and it can be applied for real-time object detection. It operates similarly to Fast R-CNN. It provides the image as an input to a convolutional network which provides a convolutional feature map. The predicted region proposals are inserted into an RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes.

R-FCN (Region-based Fully Convolutional Network) [DLHS16] increases the speed and detection performance by inserting position-sensitive score maps. It shares the computations across every single output. Each position-sensitive score map denotes one suitable location of one object class. These score maps are convolutional feature maps that have been trained to identify specific parts of each object. The R-FCN is faster than Faster R-CNN and achieves comparable performance.

YOLO (You Only Look Once) [RDGF15] comes with a different approach for analyzing the regions of interest. It uses a single CNN to predict the bounding boxes and their class probabilities. It takes an image and splits it into an NxN grid, within each part of the grid it takes M bounding boxes. YOLO is faster than Faster-RCNN, but small objects in the image hinder its performance.

SSD (Single-Shot Detector) [LAE+15] gains high-speed processing over Faster R-CNN by simultaneously performing the region proposals and region classification in two independent steps. It uses a region proposal network to generate regions of interest, then either fully-connected layers or position-sensitive convolutional layers to classify these regions, and finally predicts the bounding box and the class.

Currently, Faster R-CNN and SSD are the most widely-used object detection models. The paper [HRS+16b] presents a reliable comparison between those detection algorithms using different CNN classifiers (eg, VGG, ResNet, Inception) and concludes that Faster R-CNN is slower but more accurate than SSD.

In [LGG+17a] a variation on SSD (called RetinaNet) is presented which uses the ResNet [HZRS15] and Focal Loss [LGG+17b]. Its performance exceeds Faster R-CNN [RHGS15a], R-FCN [DLHS16], SSD [LAE+15] and YOLOv1 [RDGF15]. RetinaNet uses ResNet for feature extraction then, a Feature Pyramid Network (FPN) is used on top of ResNet to assemble a strong multi-scale feature pyramid from one single resolution input image.

In our preliminary detection experiment we used the Faster R-CNN with Inception V2 and then we used it with the RetinaNet due to its height performance. We also made a compression between these CNNs on the pedestrian detection and action recognition issues and the RetinaNet get better results, the fact that we decided

to used it in our work.

Pedestrian Detection Studies with Deep Learning

A comprehensive overview of different handcrafted feature models (classical approaches) and deep learning techniques used in the pedestrian detection task is presented in [RR19]. The study also presents various data sets available for pedestrian detection and the classical and deep learning approaches used for pedestrian detection, localization and tracking methods.

An overview of the main deep learning detection methods applied is the pedestrian detection task is presented in [BKFG19]. The authors optimize and adapt Faster R-CNN [RHGS15b], R-FCN, SSD [LAE+15], and YOLOv3 for the EuroCity Person dataset. They use Faster R-CNN, R-FCN, SSD with VGG-16 as the base classify network and YOLOv3 with the DarkNet framework and conclude that the variation of Faster R-CNN has high-grade performance on the EuroCity Person dataset [BKFG19].

A recent state of the art for pedestrian detectors based on deep learning is presented in [BFF+19]. It shows a comparison and evaluation criteria of the traditional hand-crafted features methods and Region based-CNN (R-CNN) detectors.

Pedestrian detection based on Region Proposal Network (RPN) and Boosted Forest (BF) compiled on Caltech [DWSP09], INRIA [DT05a], ETH, and KITTI [GLSU13] data sets is presented in [ZLLH16], where the method overcomes two Faster R-CNN [Gir15] limitations for pedestrian detection: unusefulness to detect the small scale pedestrians due to the insufficient resolution and the lack of a self-generating process for hard negative samples.

In [LDWW18] the authors present a pedestrian detection based on a variation of the YOLO network (where three layers were added to the original one), in order to join the shallow layer pedestrian features to the deep layer pedestrian features and connect the high, and low-resolution pedestrian features.

A variation of SSD-Inception CNN based on the SvDPed dataset is proposed in [KML+18]. The method merges the RGB images, low-resolution Lidar, and the distance between the camera and the detected object.

A deep unified model that conjointly learns feature extraction, deformation handling, occlusion handling and classification evaluated on the Caltech and ETH data sets for pedestrian detection was proposed in [OZL+17]. A solution for detecting pedestrians at different scales evaluated on the Caltech dataset by combining three CNNs, was proposed in [ESWG16]. A cascade Aggregated Channel Features detector is used in [XWK+15] to create pedestrian candidate windows followed by a CNN-based classifier for assessment purposes on monocular Caltech and stereo ETH data sets.

In [BDX16] a CNN to learn the features with an end-to-end approach is presented. This experiment focuses on the detection of small scale pedestrians on the Caltech dataset.

These algorithms are applied only RGB image modality in order to detect the pedestrians, while we also use this different image modality. Hence, we use the well known Faster RCNN and SSD detector CNNs using the default setting with RGB, Optical Flow and Depth modalities, and we also envolve out InCML model in the pedestrian detection task.

Pedestrian Action Detection Studies

Most of approaches for pedestrian action recognition are made from video data sets which are not involved in the intelligent transportation field since these fields have too few public data sets (Daimler, JAAD).

Hence, we present some related studies which are not involved in the transportation field, but the approaches are significant for the pedestrian action recognition issues, and then some related studies used in the transportation field.

Pedestrian Action Classification used in Machine Learning

A survey of pedestrian attribute recognition is presented in [WZY+19, KF18]. The authors present the current models, technical issues, action databases and the evaluation protocols.

A multi-branch classification layer for each attribute learning with a convolutional network is presented in [SSL15]. The authors assume a pre-trained AlexNet as a basic feature extraction sub-network, and replace the last fully connected layer with one loss per attribute using the KL-loss (Kullback-Leibler divergence based loss function).

A deep neural network for pedestrian attributes recognition merged with handcrafted features, and correlations between attributes is proposed in [LCH15]. The authors propose two approaches called DeepSAR and DeepMAR, where they use AlexNet as a backbone network and change the output category specified in the last dense layer into two to obtain the DeepSAR. The softmax loss is used to estimate the final classification loss. The DeepMAR runs on the human image, and its attribute label vectors concurrently and jointly estimate all the attributes through sigmoid cross entropy loss.

A joint multi-task learning algorithm for attribute estimation using CNN, called MTCNN, is introduced in [AWLJ16]. The authors assume multi-task learning to determine the corresponding attributes. The CNN models assign visual information among various attribute classes.

The algorithms, as mentioned above [SSL15, LCH15, AWLJ16], use the entire images as input and handle the multi-task learning for pedestrian actions recognition. Another approach for solving this issue is to employ local and global visual information to achieve more reliable efficiency. These algorithms use the localization of body parts, which is achieved by an external part localization module, such as part detection [LYT16, LCZH18, LHLT16], pose estimation [ZPR+13, LCZH18], poselets [BMM11], or proposal generation algorithm [JWZ13, GGM14, LLYS18, LHLT16]. This additional information improves the recognition performance significantly [BMM11, JWZ13, ZPR+13, GGM14, LYT16, LCZH18, LHLT16, LLYS18].

In [LZT+17], the authors set out to encode multi-scale features from multiple levels for pedestrian analysis using Multi-Directional Attention (MDA) modules called Attentive Deep Features for Pedestrian Analysis (HydraPlus Net). It contains the Main Net (M-Net) which is a generic CNN and Attentive Features Net (AF-Net) which involves multiple forks of multi-directional attention modules. The authors use Inception-v2 as a based network classifier.

The DIAA (Deep Imbalanced Attribute Classification using Visual Attention Aggregation) is an algorithm [SXK18] than combines the multi-scale visual attention and focal weight loss. The primary purpose is to learn the attention maps in a

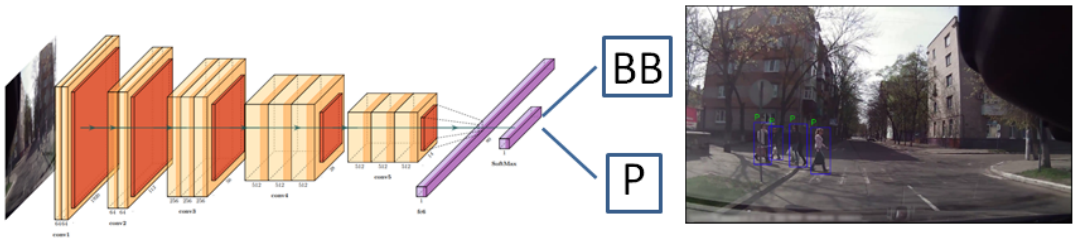


Figure 2.2 – Pedestrian detection using the same tag for all pedestrians. P=Pedestrian

weakly supervised manner to improve the classification performance by guiding the network. The authors use the features map from different layers.

Pedestrian Action Detection/Classification used in Intelligent Transportation Filed

In the transportation field, pedestrian actions recognition could be one of the potential by essential building components since the detector not only discriminates the pedestrian from other road users but also returns the pedestrian actions, the fact that could influence the car road flow. The algorithms can analyze the human body motion characteristics at a current stage of an activity using the techniques mentioned earlier or convolutional neural network.

A framework of pedestrian detection and actions recognition merged with the Euclidian distance, and joint entropy-based features selection is made in [SKA+17]. This framework was trained on the Mit, Caviar and BMW-10 datasets and tested on the MSR Action dataset, INRIA, and CASIA datasets.

A comparative study on recursive Bayesian filters combined with Extended Kalman Filters (EKF) and Interacting Multiple Models (IMM) performed on four pedestrian motions (crossing, stopping, bending in, starting) is presented in [SG13a]. The method performed on the Daimler dataset [SG13a].

A pedestrian action recognition method using the width and height of the bounding box and centroid of human shape to determine the pose ratio based on the Daimler dataset is presented in [HK16]. The method merges the pose ratio with pedestrian walking speed, and direction and a spatial plan of the background in order to perform action recognition.

A detection of pedestrians crossing the road is presented in [HJ15a]. The authors use the optical flow to detect the movement of pedestrians, then use the KLT tracker to find the corresponding features in progressive images followed by a classification step which classifies each block into a motion region and finally a probabilistic generation of the foreground mask is applied, to find if the pedestrian is crossing or not the street. The authors use the correlation of the pedestrian ratio for the width and height of the detected bounding box and the ratio of the centroid position from the ground level divided by the height of bounding box, to carry out the action recognition task.

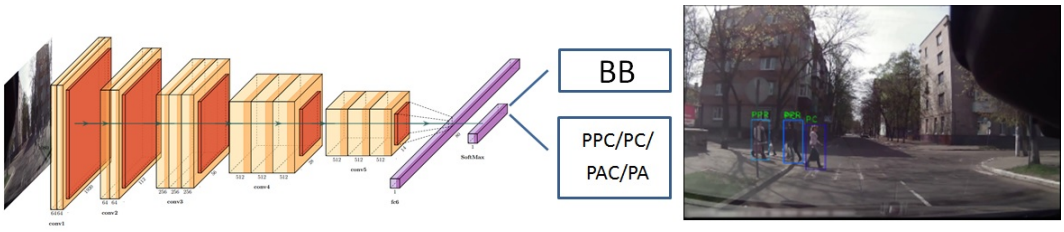


Figure 2.3 – Pedestrian detection using multiple tags. PPC: Pedestrian is Preparing to Cross the street; PC: Pedestrian is Crossing the street, PAC: Pedestrian is About to Cross the street; PA: Pedestrian intention is Ambiguous (PA)

2.3 Pedestrian Detection

In this section, we present our pedestrian detection component followed by the steps that we use to extract the Depth image modality and Optical Flow image modality from JAAD [KRT16] dataset in order to use them in our Incremental Cross Modality Deep Learning approach descried in the first chapter.

2.3.1 Pedestrian Detection Component

In order to develop a pedestrian detection system it is mandatory to take into account three main components: the sensors employed to capture the visual road environment, the processing elements, and the classification parts. In general, all these components has to be together developed to achieve a high detection performance, but seldom are specific items that could be investigated independently according to the target application. We have examined the detection part by applying a generic object detector based on the public RetineNet [LGG+17a] and Faster R-CNN [RHGS15a]. We have handled the Resnet50 CNN [HZRS15] and Inception V2 CNN [HRS+16a] architectures for the classification task with the Keras public open source RetinaNet implementation described in [LGG+17a] and with TensorFlow public open source Faster R-CNN Inception V2 implementation described in [HRS+16a]. All the training process are based on the JAAD [RKT17a] dataset, that provides an annotation of pedestrians with behavioral tags and pedestrians without behavior tags. We focus on finding whether the pedestrian is crossing or not crossing the street at the current time (t=0) and the pedestrian detector is the first component. The Jaad dataset [RKT17b, RKT18] descriptions and annotations present various specific events and actions made by pedestrians before crossing the street, thus we divide the pedestrian actions into four classes: pedestrian is preparing to cross the street (PPC), pedestrian is crossing the street (PC), pedestrian is about to cross the street (PAC), and pedestrian intention is ambiguous (PA). Moreover, the JAAD dataset provides a unique id for each pedestrian in a given video.

We adopted four approaches for the training stage:

1. using all pedestrian samples where we consider all the annotation tags as a pedestrian (P) (see Fig 2.2);

2. using the four proposed pedestrian tags mentioned above and taking into account only the pedestrian behaviors (PPC, PC, PAC, PA) (see Fig 2.3).

3. using the four proposed pedestrian tags mentioned above and taking into account both the pedestrian behaviors and the id according to JAAD [KRT16]

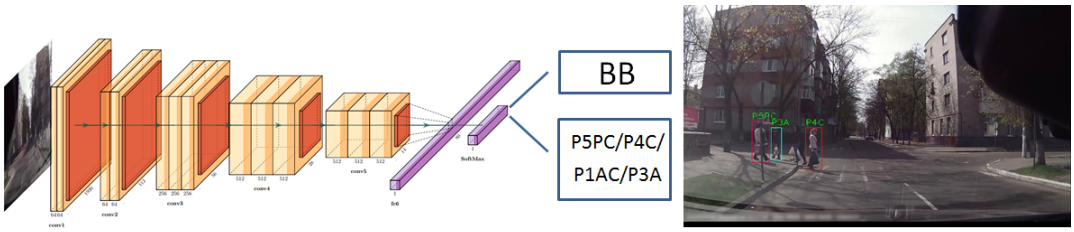


Figure 2.4 – Pedestrian detection using pedestrian actions and id. e.g. pedestrian1 cross= P1C, pedestrian2 preparing to cross=P2PC, pedestrian3 ambiguous=P3A

pedestrian annotations (e.g. pedestrian1 cross= P1C, pedestrian2 cross=P2C, pedestrian3 ambiguous=P3A) (see Fig 2.4).

4. applying the Incremental Cross Modality Learning (InCML) using RGB, Depth and Optical Flow (described in Chapter 1) on (1) and (2) approaches mentioned below (see Fig 2.5).

The JAAD dataset offers only the RGB image modality. In order to apply the InCML, we have to extract the Depth and Optical Flow image modality. In the this chapter, we use RGB image modality instead of Intensity in the InCML approach (see Fig 2.5).

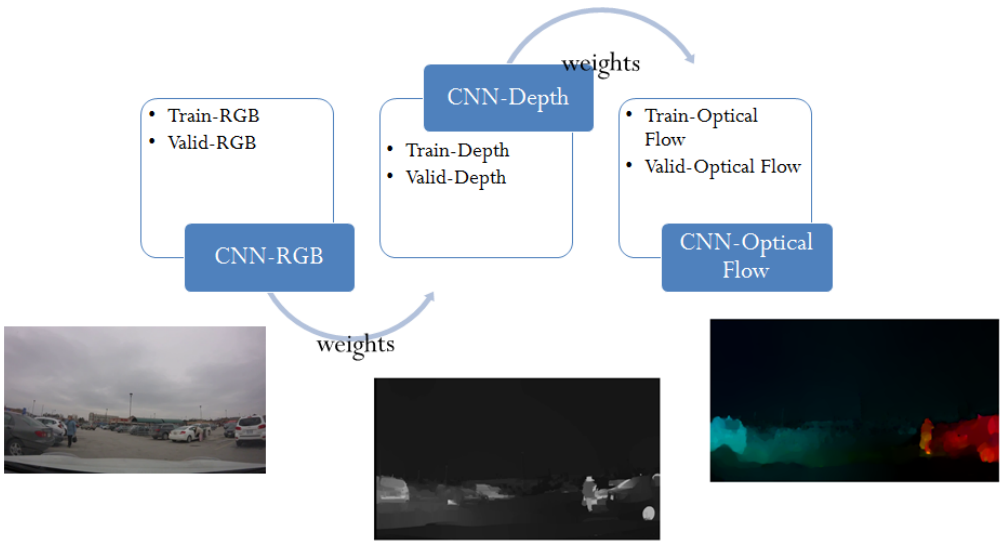


Figure 2.5 – The InCML Pedestrian Detection Architecture

2.3.2 Depth Modality from JAAD Dataset

In order to obtain the Depth image modality, we use the public source code provided by OpenCV, which is based on the equivalent triangles method. The algorithm is inversely proportional to the difference in distance of corresponding image points and their camera centers.

$$\text{disparity} = x - x' = \frac{\mathrm{B}f}{\mathrm{Z}} \tag{2.1}$$

In this formula x and x' express the distance between points in the image plane, B represents the gap between two cameras and f signifies the focal length of the camera. This values are already known and are set by default, but with the possibility of

changes. We use the default values since we do not have available this information. On other words, the equation mentioned above states that the depth of a point in a scene is inversely proportional to the difference in distance of corresponding image points and their camera centers. Therefore with this information, we can derive the depth of all pixels in an image, which it finds corresponding matches between two images.

We obtain the Depth image modality for the JAAD dataset by applying the code mentioned above on the frame at times T and T-1 time (see Fig 2.6). We adjusted the values of the number of Disparities at 32 and blockSize at 10 that helped us to get better results.

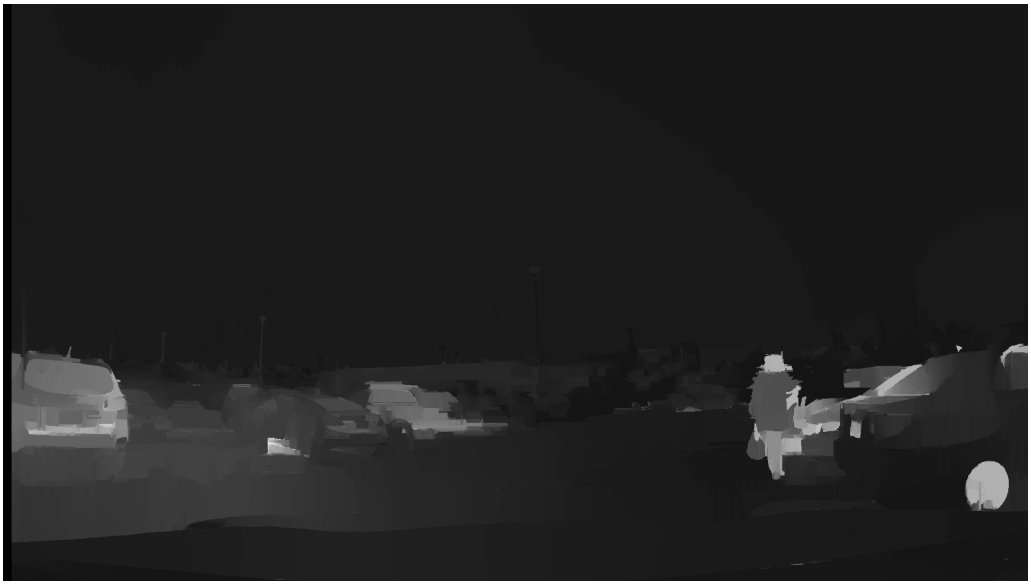


Figure 2.6 – The JAAD Depth image modality sample

2.3.3 Optical Flow Modality from JAAD Dataset

In order to extract the Optical Flow from JAAD video data sets, we used the public open source code "Python Dense Optical Flow," [1] code which was initially developed as part of the [PGD+17] project. It is a fast and reliable optical flow method based on the Coarse2Fine warping method from Thomas Brox. It uses the conjugate gradient for solving large linear systems rather than Gauss-Seidel or SOR like other public source codes. The conjugate gradient provides a matrix A which has the concatenations decomposed of filtering and weighting. We obtain the optical flow for the JAAD dataset by applying the code mentioned above on the frame at time T and T-1 (see Fig 2.7).

[1] https://github.com/pathak22/pyflow

Figure 2.7 – The JAAD Optical Flow image modality sample

2.4 Experiments

In this section, we present our set of experiments including setups and performance assessment of our approaches.

2.4.1 Data setup

There are many different large-scale pedestrian detection data sets used in computer vision and deep learning and mainly designed for automotive safety issue. The main difference comes from the number of samples, image size, the way how the data were captured (monocular, stereo, infrared, among others), the weather condition, the context of the date and also what additional information provides about pedestrian like bounding box, pedestrian behavior, pedestrian action, pedestrian ids etc. We highlight only a few of them which we consider that are the most important ones, and we summarized the differences between them in the Table 2.1.

The purpose of the thesis is concerned with pedestrian action recognition and pedestrian tracking for traffic collision risk assessment, more than pedestrian detection, which is a preliminary step, we naturally decided to use the JAAD dataset [RKT17a] in our experiments.

The experiments were performed on the JAAD dataset [RKT17a] because its data

Table 2.1 – Comparison of the pedestrian detection data sets.

Dataset	KITTI [GLSU13]	Caltech [DWSP09]	MPD [HPK+15]	INRIA [DT05a]	Daimler Mono [EG09]	JAAD [KRT16]	ESP [BKFG19]
No ped. samples	12k	347k	86.2k	1.8k	72k	337k	238k
No frames	80k	250k	95k	2.5k	28.5k	82k	47k
OCC. Labels	x	x	x			x	x
Temporal Corr.		x	x			x	x
Video sequences	x	x	x		x	x	x
Behavior data						x	
Context data						x	
Weather Variation						x	x
Geographical Variational						x	x

was collected in usual urban road traffic environments for different locations, times of the day, road and weather conditions. This dataset provides pedestrian bounding boxes (BB) for pedestrian detection (including, for several of them, the pedestrian actions), pedestrian attributes for estimating the pedestrian behavior and traffic scene elements. It has 346 video sequences (between 5 and 15 seconds long) with an image resolution of 1920 x 1080 and respectively 1280 x 720 pixels recorded in different urban environments [KRT16]. Moreover, it contains approximately 337k pedestrian samples of which approximately 72.000 (18%) samples are tagged as partially occluded BBs and 46000 (11%) samples as heavily occluded BBs.

We use all the pedestrian samples, including the partially and heavily occluded pedestrians, for all training and testing processes.

2.4.2 Training protocol

We used the first 250 video sequences for the training process and the rest for testing. The training and testing samples include also partially occluded and heavily occluded BBs.

In [RKT17b, RKT18], the authors present a variety of pedestrian behaviors done before crossing and after crossing the street and even when the pedestrian does not cross the street. These behaviors were collected and annotated with different action labels according to the pedestrian events for each pedestrian from all the video sequences.

The events could be:

- the pedestrian completes to cross the street;
- the pedestrian has no intention to cross the road (e.g. sits on a public bench, waiting for public transportation);
- the pedestrian does not cross the street (e.g the pedestrian has started to cross the street but suddenly he/she is stopping).

For instance, if the pedestrian is going to cross the street, he/she can do a minimum of actions like standing, looking, and then crossing the street, or moving, looking, and then crossing the street. The pedestrian actions applied before or during one event could be different for each pedestrian, even if the event is the same. Hence, according to these action annotations, we can observe there exists a typical pattern of actions timeline/succession for each pedestrian for each event (see Figure 2.8).

Therefore, according to the specifications and annotations presented above, we divide the pedestrian labels into four classes:

1. Pedestrian is Preparing to Cross the street (PPC), where the pedestrian is walking/standing, paying attention or not and changing or not its behavior before crossing. In this case, the actions could be: moving, looking, standing, nodding, glancing, hand waving, slowing down, and finally crossing the street. We take into account all the actions up to the crossing event as being in the PPC class. In this case, the pedestrian were definitely crossing the street after these actions.

2. Pedestrian is Crossing the street (PC), where the pedestrian is observed from the point of crossing until he/she has crossed the road. In this case, it is

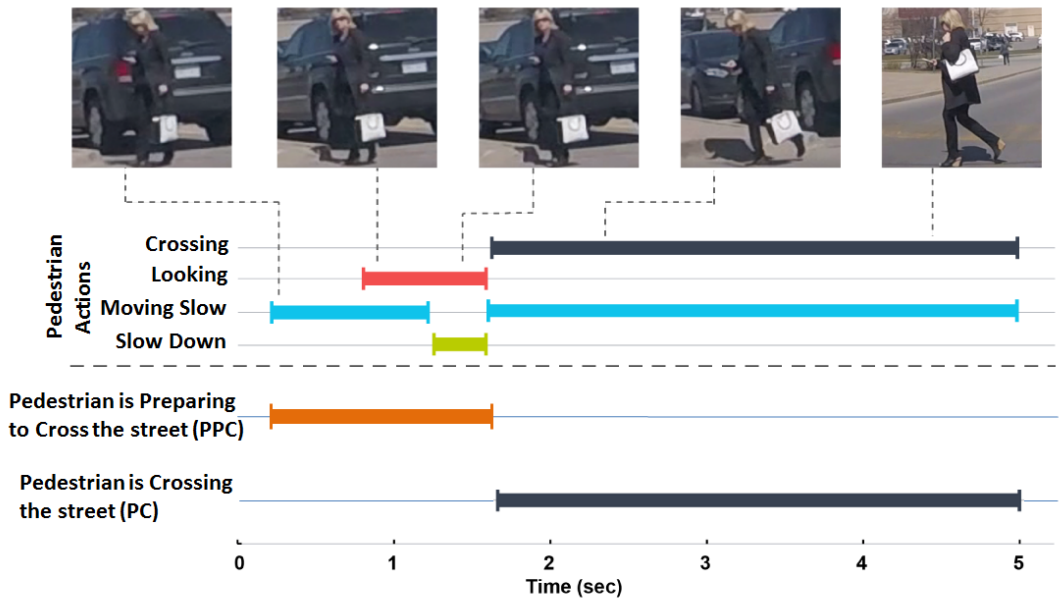


Figure 2.8 – Timeline of events/actions whenever the pedestrian is going to cross the street. This image was picked from JAAD [KRT16] dataset source and modified/updated to our requirements.

mandatory to have a crossing action during this event but is not mandatory to have a specific event before the crossing event. There are video sequences where the pedestrians are annotated only from point of crossing the street. The pedestrian behavior could involve other actions like looking, hand waving, speeding up, nodding, slowing down, glancing during this event.

3. Pedestrian is About to Cross the street (PAC), where the pedestrian is about to cross and pays attention and responds according to the event. In this case, the actions could be: moving, looking, standing, noding, glancing, hand waving, slowing down, but the pedestrian will not crossing the street. The pedestrian is definitely not crossing the street after these actions.

4. Pedestrian intention is Ambiguous (PA), where the pedestrian is walking or standing, and his/her intention is ambiguous. In this case, the actions could be: moving, looking, standing, glancing, speeding up. We consider all the actions after the pedestrian crosses the street. In this case, the pedestrian has crossed the road or other events which do not present a risk situation.

2.4.3 The convolution neural network setups

We train the Convolution Neural Network (CNN) in three ways:

1. We train the CNN with all pedestrian samples where we consider all the annotation tags as a pedestrian (P);

2. We train the CNN with the pedestrian action tags mentioned above (PPC, PC, PAC, PA);

3. We train the CNN with the pedestrian actions tags (PPC, PC, PAC, PA) with the pedestrian's id for each pedestrian/video sequence according to JAAD pedestrian id annotations.

In this work we have to analyzed whether pedestrians cross and pedestrian do not cross the street. Hence in the first 250 video sequences (using the original resolution, 1920x1080) we used for the training process. We have 24324 samples of pedestrians who are preparing to cross the street (PPC), 51012 samples where pedestrians are crossing the street (PC), 14267 samples where pedestrians are about to cross the street (PAC) and 5567 samples where th pedestrian's intentions are ambiguous (PA).

The pedestrian identifications were independently made for each pedestrian and each video sequence [KRT16]. Hence, we have 13 pedestrian ids for pedestrians who are preparing to cross the street (PPC), 13 ids for pedestrian which are crossing the street (PC), 12 ids for pedestrians who are about to cross the street (PAC) and 11 ids for pedestrians whose intentions are ambiguous (PA).

We perform the CNN learning process during 48 hours on 2 GPU, with a batch size of 1, using an initial learning rate value of 0.0005 with ADAM algorithm learning.

2.4.4 Testing protocol

The testing set used to assess the CNN model performances is independent of the training dataset. It contains 105 video sequences. It has a total of 43420 samples, where 12110 examples are pedestrians who are preparing to cross the street (PPC), 19157 samples are pedestrians who are crossing the street (PC), 1296 samples are pedestrians who are about to cross the street (PAC) and 4857 examples where their intention is ambiguous (PA).

The testing methodology consists in the following:

- testing independently the pedestrian detection component for each image modality: RGB, Depth and Optical Flow;
- testing the pedestrian action detection;
- testing the Incremental Cross Modality pedestrian detection;

2.4.5 Evaluation protocol

The evaluation process for all the CNN models was done with Tensorflow Deep Neural Network Framework. The performances are assessed by the average precision (AP) and mean average precision (mAP) for the detection part. The AP and mAP values were computed using the TensorFlow metrics tool. The AP is calculated as the area under the curve (AUC) of the Precision x Recall curve. It is the precision averaged across all recall values between 0 and 1.

$$\text{Precision} = \frac{\text{True Positive}}{\text{All prediction detection}} \tag{2.2}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{All ground truths}} \tag{2.3}$$

$$\mathrm{AP} = \frac{1}{11} \sum_{r=\{0,0.1,....1\}} \rho_{interp(r)} \quad \text{with} \quad \rho_{interp(r)} = \max_{\tilde{r}:\tilde{r}\geq r} \rho(\tilde{r}) \tag{2.4}$$

The $\rho(\tilde{r})$ is the measured precision at recall $\tilde{r}$.

Instead of using the precision observed at each point, the AP is obtained by interpolating the precision only at the 11 levels taking the maximum precision whose recall value is greater than r.

The AP is calculated only for each class, which has the detection-result higher than 50% (Intersection over Union, IoU ≥ 0.50).

Intersection Over Union (IoU) estimates the overlap between the ground truth bounding box (BB_{gr}) and the predicted bounding box (BB_{pr}). The IoU presents whether a detection is correct (True Positive, IoU ≥ 0.5) or wrong (False Positive, IoU < 0.5).

$$\mathrm{I}o\mathrm{U} = \frac{area(\mathrm{BB}_{gr} \cap \mathrm{BB}_{pr})}{area(\mathrm{BB}_{gt} \cup \mathrm{BB}_{pr})} \tag{2.5}$$

We calculate the margin of error (Confidence Interval - CI at 95 % confidence) to evaluate whether one model is statistically better than another one.

$$\mathrm{CI} = 1.96\sqrt{\frac{\mathrm{P}(100-\mathrm{P})}{\mathrm{N}}}\%. \tag{2.6}$$

In this formulation, P represents the performance system (e.g., AP, mAP) and N represents the number of testing samples.

2.5 Evaluation and Results

The experiments were performed on the JAAD dataset (on the original video size) which provides only the RGB image modality. We extracted the Optical Flow and Depth image modality in order to apply our Incremental Cross-modality learning depicted in Chapter 1. We independently provide the results for the pedestrian detection component for each image modality: RGB, Depth and Optical Flow (using the classical deep learning approach), followed by the pedestrian action results, then the results for the Incremental Deep Learning approach applied on the pedestrian detection component and pedestrian action component.

2.5.1 Evaluation of the Uni-Modal Pedestrian Detection Component

In order to test the detection performance, we carried out several experiments.

Table 2.2 – Our detection performances using one label. One label represents that all samples are tagged only as a pedestrian (without action recognition).

Approach	Learning on	Testing on	mAP±CI
Classical Unimodal	RGB	RGB	56.05±0.93
	Optical Flow	Optical Flow	53.12±0.91
	Depth	Depth	46.5±0.85

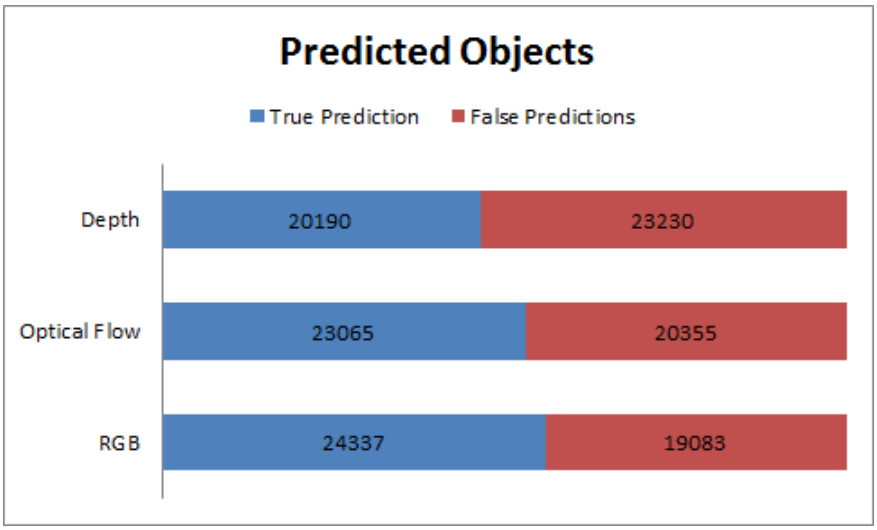


Figure 2.9 – The Uni-modal true positive detection performance on each image modalities.

Our detection results are summarized in Table 2.2. We observed that the detection performance obtained with the classical approach (using all samples as pedestrians) gives a good performance on the JAAD dataset since it has to distinguish the pedestrian from other road users.

We obtained mAP=56.05% on the RGB modality followed by the Optical Flow modality with mAP = 53.12% and finally the mAP = 46.50% for Depth modality. According the Figure 2.9, from 43420 testing samples, the classical detection approach managed to detect 24437 RGB samples, 23139 Optical Flow samples, and 20190 Depth samples. We observe that the detection performance between RGB image modality and Optical Flow are quite close. Based on this fact, we assume that data which were obtained from different locations, times of the day, roads and weather conditions affect the detection component according to each image modality.

2.5.2 Evaluation on Uni-Modal Incremental Cross-Modality Deep Learning Pedestrian Detection

Since the incremental cross-modality learning (InCML) method is the most promising approach in Chapter 1, we decided to carry out more extensive experiments. Thus, the InCML model was applied to the detection approach using RetineNet [LGG+17a] on each image modality, RGB, Optical Flow, Depth. We kept the same learning order as in Chapter 1 but instead of Intensity image modality, we used RBG image modality. Hence, the RGB learning order in InCML approach starts with Depth motility followed by Optical Flow and then RGB image modality. The Optical Flow learning order for InCML is RGB, followed by Depth images and finally, Optical Flow image modality. The Depth learning order InCML is RGB, then Optical Flow and, finally, Depth image modality.

The detection performance is presented in Table 2.3 and we obtained mAP= 59.65% on the RGB modality followed by the Optical Flow modality with mAP = 53.61% and finally the mAP = 45.28% for Depth modality. The InCML outperformed the classical uni-modal detection approach on the RGB and Optical Flow image modality, but its performance is statistically significant only for the RGB image modality Δ_{RGB}=1.70.

Table 2.3 – Our Classical vs InCML detection performances using one label. One label represents that all samples are tagged only as a pedestrian.

Approach	Learning on	Testing on	mAP±CI
Classical Unimodal	RGB	RGB	56.05±0.93
	Optical Flow	Optical Flow	53.12±0.91
	Depth	Depth	46.5±0.85
InCML	Depth_i+Optical Flow_i+RGB_i i=$\overline{1,n}$	RGB	59.65±0.96
	RGB_i+Depth_i+Optical Flow_i i=$\overline{1,n}$	Optical Flow	53.61±0.91
	RGB_i+Optical Flow_i+Depth_i i=$\overline{1,n}$	Depth	45.28±0.84

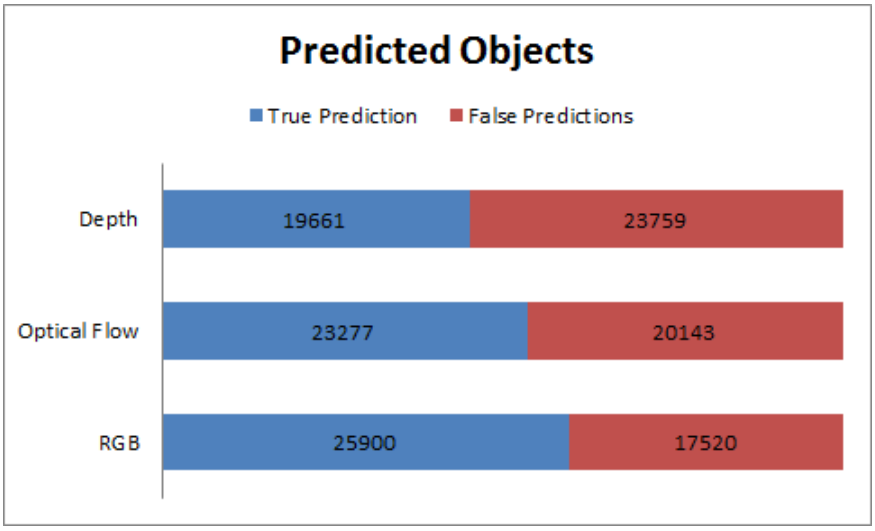


Figure 2.10 – The Incremental Cross Modal true positive detection performance.

In Fig 2.10, we present a comparison between true detection and false detection. We observed the significant accurate detection prediction is achieved only with the InCML RGB modality since the performance obtained with the Optical flow is better but not statistically significant. The highlight comes from the Depth image modality, where its performance is worse than classical uni-modal. Although the difference is not significant, we believe that this degradation result because we have derived the Depth modality from mono camera videos.

The previous CNN directly influences this method because it is based on transfer learning. The transfer learning takes the weight information from the previous CNN, which has already been trained. The InCML approach achieves better results due to this transfer learning technique. In the second and third learning steps, the InCML takes more specific and optimal weight information according to the target application. For instance, our target application is to detect pedestrians. If we use the weight information from a CNN whose is target application was to detect 1000 objects, we will find a lot of information about the purpose (e.g., cars, plants, animals) which does not correspond to the pedestrian detection target application. The first CNN in the InCML approach is focused on taking into account only the information required for the target application. Then the second and third will optimize only the data specific to its purpose.

Table 2.4 – The comparison between Faster R-CNN Inception V2 and RetinaNet using the classical unimodal approach on JAAD dataset. The labels represent: PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC= Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous.

CNN	Learning on	Testing On	PC	PPC	PAC	PA	mAP
			AP				
Faster R-CNN Inception V2	RGB	RGB	64.99	13.65	11.01	9.63	24.82
RetinaNet	RGB	RGB	65.57	17.67	13	9.22	26.36

However, this method is a generic model. It is more flexible, allowing for adaptive settings according to each CNN detector. It could be used and adapted for all types of CNN, not only for a specific one.

2.5.3 Evaluation of Uni Modal Pedestrian Action Detection

In our first experiment, we made an analogy between RetinaNet [LGG+17a] and Faster R-CNN-Inception v2 [ZLLH16] performances. The experiment was carried out on the RGB image modality using multiple pedestrian tags, PPC, PC, PAC, PA. We observed the RetineNet returned better performance than Faster R-CNN Inception v2 except for a PA case where the Faster R-CNN Inception v2 achieved a better result (9.63 AP), but its performance is not statistically significant. The comparison between RetinaNet and Faster R-CNN performance is summarized in Table 2.4. Since the RetinaNet returned a better result than Faster R-CNN Inception v2, we decided to use it in the next experiments.

Table 2.5 – Our detection performances using multiple output labels. The labels represent: PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC= Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous.

Approach	Learning on	Testing on	PC	PPC	PAC	PA	mAP
			AP	AP	AP	AP	±CI
Classical Unimodal	RGB	RGB	65.57 ±1.35	17.67 ±1.36	13 ±1.54	9.22 ±1.63	26.36 ±0.83
	Optical Flow	Optical Flow	62.87 ±1.37	14.74 ±1.26	1.00 ±0.46	8.89 ±1.60	24.13 ±0.80
	Depth	Depth	52.34 ±1.41	9.32 ±1.04	2.55 ±0.72	7.08 ±1.44	17.82 ±0.72

The detection performance using RetinaNet approach (using multiple pedestrian tags, PPC, PC, PAC, PA) is presented in Table 2.5. We observed that the detection performance is worse than the classical pedestrian detection since it has to distinguish the pedestrian from other road users and even its actions.

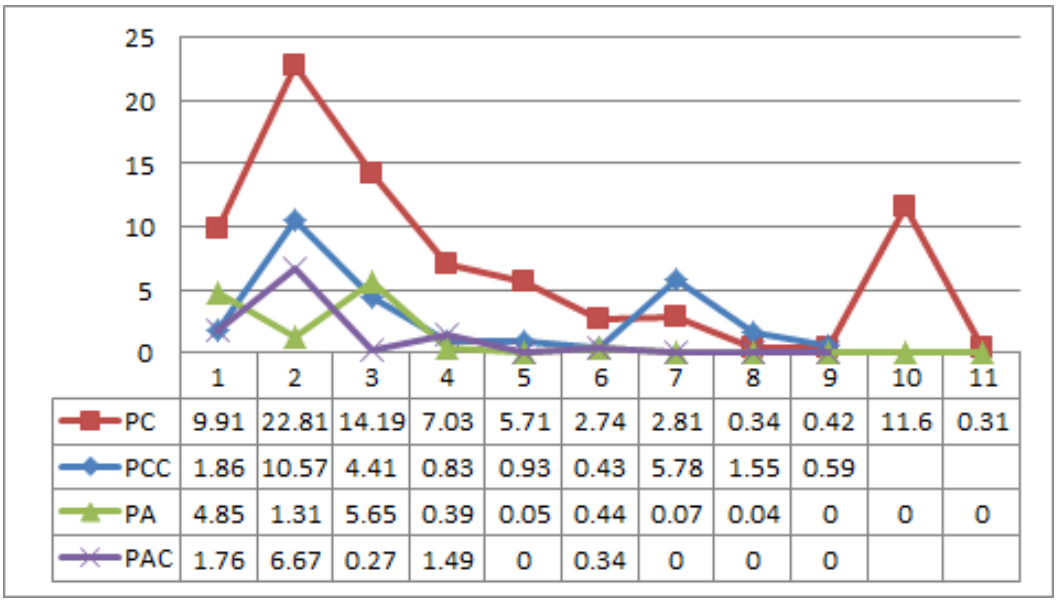
We archived 26.36 mAP using the RGB modality then 24.13 mAP using Optical Flow and finally 17.87 mAP using the Depth modality.

The main objective of this approach is to find out if a pedestrian is crossing, or whether the pedestrians action presents a critical situation. The most crucial cause for the pedestrian and drivers is when the pedestrian is crossing, and the car cannot stop or avoid it in time.

According to Table 2.5 and Figure 2.15 we observe that the detection performance when a pedestrian is crossing the street (PC) is the highest one for all image modalities (65.57% AP for RGB, 62.87% AP for Optical Flow, 52.34% for Depth modality).

This leads us to believe that if a pedestrian detection system returns even the pedestrians current action, the vehicle can act according to the situation and also to avoid a collision.

The PC detection's performance is followed by a pedestrian who is preparing to cross the street (PPC) (17.67% AP for RGB, 14.74% AP for Optical Flow, 9.32% for Depth modality),a pedestrian is about to cross the street (PAC) (13.00% AP for RGB, 1.00% AP for Optical Flow, 2.55% for Depth modality), and finally the pedestrian's intention is ambiguous (PA) (9.22% AP for RGB, 8.89% AP for Optical Flow, 7.08% for Depth modality). The highlight is in the PAC situation, where the Depth modality achieved better performance than Optical Flow. We deem the Optical Flow performance for the PAC situation is less than the Depth modality because PAC and PPC configurations are too closer, impossible even for a human being to distinguish between them. We have evaluated that the pedestrian action detection takes into



	1	2	3	4	5	6	7	8	9	10	11
PC	9.91	22.81	14.19	7.03	5.71	2.74	2.81	0.34	0.42	11.6	0.31
PCC	1.86	10.57	4.41	0.83	0.93	0.43	5.78	1.55	0.59		
PA	4.85	1.31	5.65	0.39	0.05	0.44	0.07	0.04	0	0	0
PAC	1.76	6.67	0.27	1.49	0	0.34	0	0	0		

Figure 2.11 – Pedestrian Actions detection performance using ids. The horizontal values represent the pedestrian ids. The Vertical values represent the AP performances. PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC=Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous.

account even the pedestrian id according to JAAD annotations in order to unify the deep learning step with tracking the pedestrians step (see Figure 2.16). Each pedestrian has a unique identifier on each video. For this method, we analyzed only on the RGB image modality. Since its performance is too low (Figures 2.11 and 2.14), we decided not to continue to experiment with other image modalities and detection methods. We deem it is a better solution to apply a separate tracking algorithm or to find a new solution for fusing the deep learning with the tracking issue (see Chapter 3).

2.5.4 Evaluation of Incremental Cross Modality Deep Learnig Pedestrian Action Detection

We evaluate the pedestrian action detection using the incremental cross-modality learning (InCML) since it is the most promising approach in Chapter 1.

We keep the same learning order as in Chapter 1 and in the previous InCML detection method but instead of the Intensity image modality, we used the RBG image modality.

The detection achievement is shown in Table 2.6. We obtained mAP=29.54% on the RGB modality followed by the Optical Flow modality with mAP = 22.30% and finally the mAP = 18.17% for Depth modality. The InCML outperformed the classical uni-modal pedestrian action detection approach on all image modalities, but its performance is statistically significant only for the RGB image modality Δ_{RGB}=1.63.

The detection performance (see Table 2.6 and Fig 2.15) if a pedestrian is crossing the street (PC) is the highest one for all image modalities (68.68% AP for RGB, 62.68% AP for Optical Flow, 53.35% for Depth modality), followed by pedestrian is preparing to cross the street (PPC) (21.72% AP for RGB, 15.26% AP for Optical Flow, 10.60% for Depth modality), followed by pedestrian is about to cross the street (PAC) (15.93% AP for RGB, 10.60% AP for Optical Flow, 4.33% for Depth modality), and finally pedestrian intention is ambiguous (PA) (11.82% AP for RGB, 9.90% AP for Optical Flow, 6.20% for Depth modality).

We observe that the performance of the InCML detector is directly proportional to the performances of each pedestrian action detection.

Table 2.6 – Our Classical vS InCML detection performances using multiple output labels. The labels represent: PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC= Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous.

Approach	Learning on	Testing on	PC	PPC	PAC	PA	mAP
			AP	AP	AP	AP	±CI
Classical Unimodal	RGB	RGB	65.57 ±1.35	17.67 ±1.36	13 ±1.54	9.22 ±1.63	26.36 ±0.83
	Optical Flow	Optical Flow	62.87 ±1.37	14.74 ±1.26	1.00 ±0.46	8.89 ±1.60	24.13 ±0.80
	Depth	Depth	52.34 ±1.41	9.32 ±1.04	2.55 ±0.72	7.08 ±1.44	17.82 ±0.72
InCML	RGB	RGB	68.68 ±1.31	21.72 ±1.47	15.93 ±1.68	11.82 ±1.82	29.54 ±0.86
	Optical Flow	Optical Flow	61.68 ±1.37	15.26 ±1.28	10.60 ±1.41	9.90 ±1.68	24.61 ±0.81
	Depth	Depth	53.35 ±1.41	9.79 ±1.06	4.33 ±0.93	6.2 ±1.36	18.17 ±0.73

2.5.5 Comparison of the Uni-Modal vs Incremental Cross Modality Deep Learning Pedestrian Detection for Pedestrian Action Detection

The comparison between Uni-Modal Pedestrian Detection and Incremental Pedestrian Detection was made independently for each situation:

- Comparison of Uni-Modal Pedestrian Detection and Incremental Pedestrian Detection using only one class (the first detection approach where the classes is only Pedestrian=P) and using multiple class (the second approach, where the classes are PC, PPC, PAC, PA);
- Comparison of pedestrian action detection for each imaging modality.

Comparison of Uni-Modal Pedestrian Detection and Incremental Cross Modality Deep Learning Pedestrian Detection

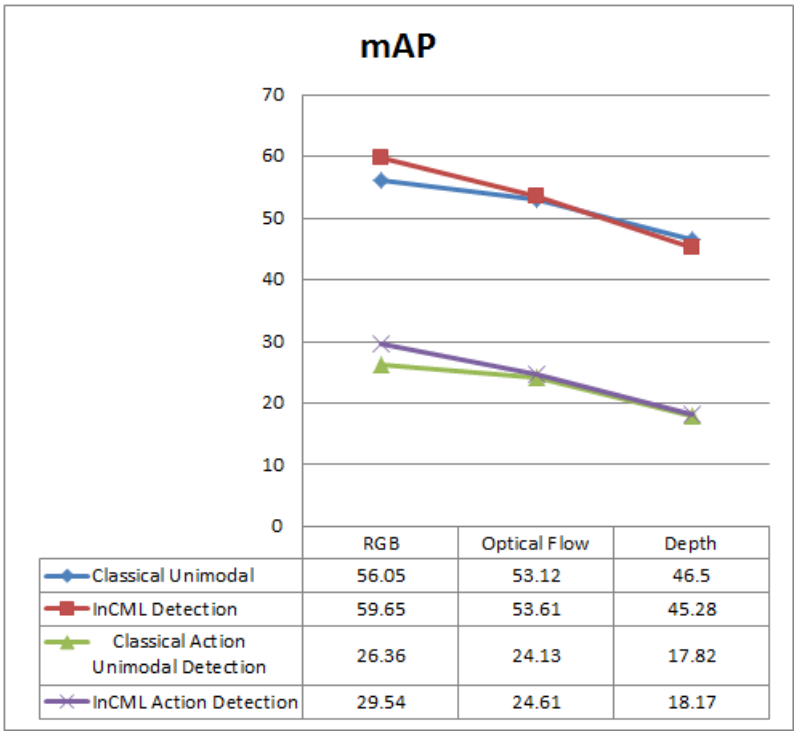


Figure 2.12 – Comparison of Uni Modal Pedestrian Detection and Incremental Cross-Modality Pedestrian Detection.

We observed from Fig 2.12 that the detection performance achieved with the classical approach (using all samples as pedestrians) performs well on the JAAD dataset since it has to identify the pedestrian from other road users. The second approach (using multiple pedestrian tags), although it detects the pedestrians, cannot be associated with the first method because it also instantly classifies the action of the pedestrian during the detection step. Therefore its performance is lower than the first classical detection approach. On the other hand, pedestrian detection using the multiple tags approach could be a starting point for a deep investigation. This approach estimates the pedestrian actions at the current time (T=0) and could be beneficial for developing a pedestrian prediction system. We can not

compare our detection models with JAAD approaches [RKT17a] as our results are not directly comparable. The authors made a classification for a specific pedestrian action based on pedestrian attention information and used the only non-occluded pedestrian samples [RKT17a]. Their approach is based on a variation of the AlexNet-Imagenet CNN, where the input data are cropped beforehand.

For the majority of the image modalities, the InCML approach outperformed the classical uni-modal detection for both patterns improvement with and without action classification. In the first approach, the InCML obtained statistically significant performance only for the RGB image modality (Δ_{RGB}=1.70) except for the Depth image modality; the return is less than the classical approach.

In the second approach, InCML exceeded the classical uni-modal detection for all image modalities, but the only notable performance was yielded for the RGB image modality (Δ_{RGB}=1.63).

We conclude that the InCML detection method achieved statistically significant results only for the RGB image modality because the JAAD dataset only offers monovision RGB videos. We consider that to acquire high-grade performance for the depth and optical flow, we require a more optimal and reliable algorithm to derive those image modalities from monovision.

Comparison of Pedestrian Action Detection for each Imaging modality

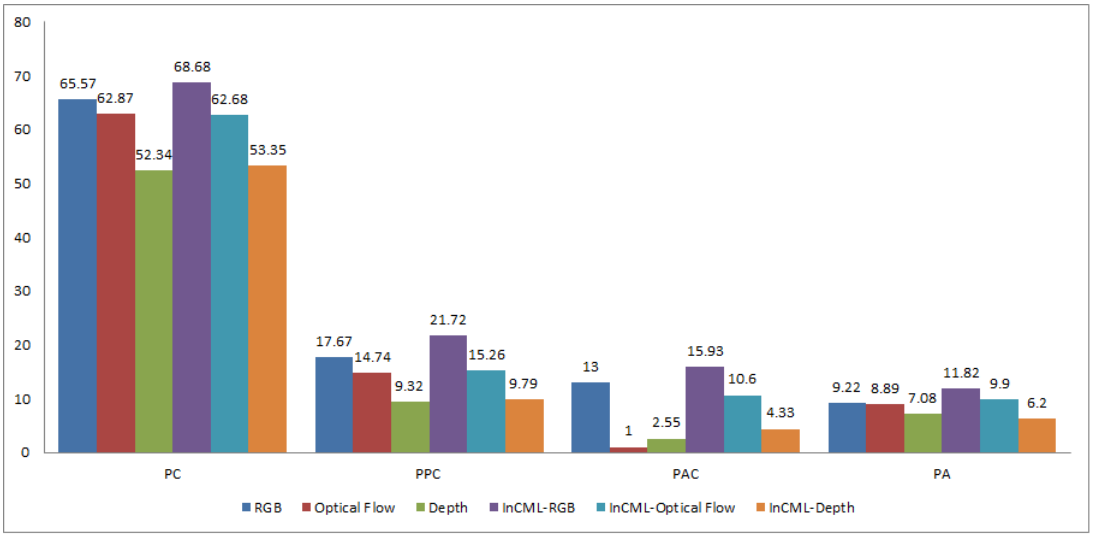


Figure 2.13 – Comparison of Pedestrian Action Detection for each Imaging modality.

We observe from Fig 2.13 that the InCML pedestrian action detection using the RGB image modality exceeds all detection modalities. Its achievement is statistically significant when the pedestrian is crossing the street (ΔPC_{RGB}=0.34) followed by pedestrians who are preparing to cross the street (ΔPCC_{RGB}=1.66) and finally the pedestrians who are about to cross the street (ΔPAC_{RGB}=0.66). When the pedestrian intention is ambiguous, the InCML outperforms the classical uni-modal approach, but the performance is not statistically significant.

The InCML results using the Optical Flow image modality exceeds the classical unimodal approach for PPC, PAD, and PA situations. Its performance is statistically notable only for PAC (ΔPAC_{RGB}=7.46). When the pedestrian is crossing the street, the InCML achieved slightly lower performances than the classical unimodal approach.

The InCML performance using the Depth image modality slight exceeds the classical unimodal approach for PC, PPC, and PAC situations. Its performance is not statistically significant. When the pedestrian intention is ambiguous, the InCML achieved slightly lower performances than the classical unimodal approach.

We consider that the dispersion of results is due to the power of optimization and discrimination of the InCML method.

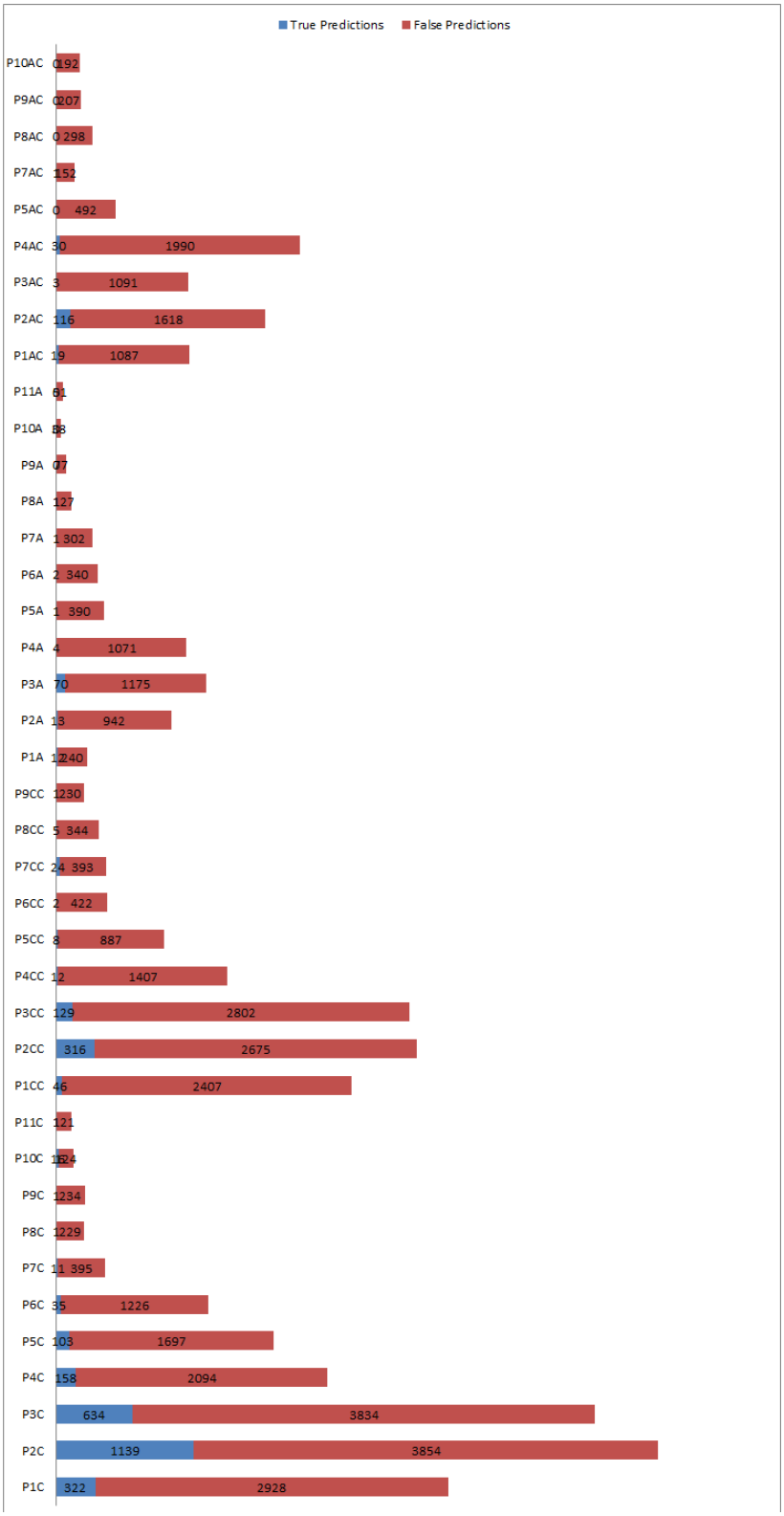
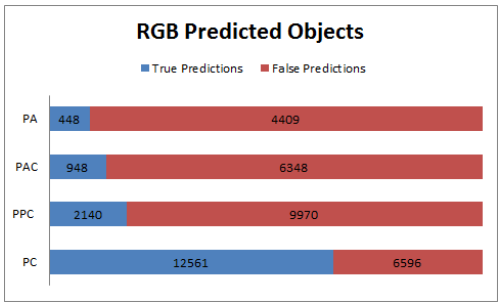
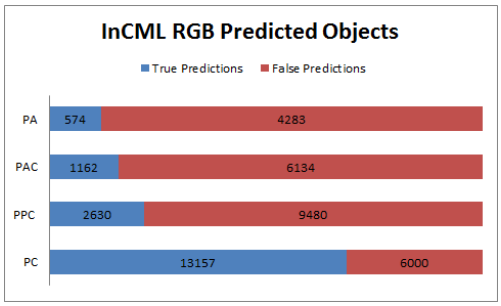


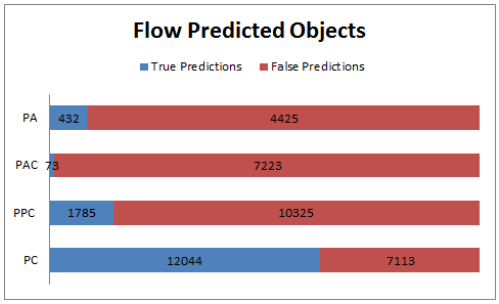
Figure 2.14 – Pedestrian Action Detection performance using ids. The horizontal values represent the pedestrian ids. The Vertical values represent the AP performances. PPC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC=Pedestrian is About to Cross the street and PA= Pedestrian intention is Ambiguous.



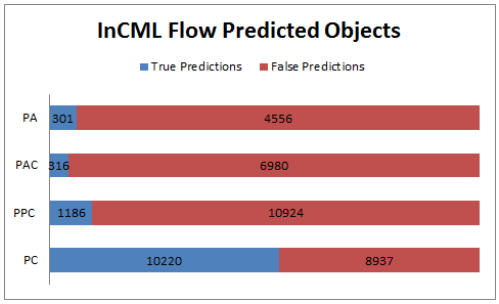
(a) Predicted Objects performance using Uni-Modal Pedestrian for RGB image modality



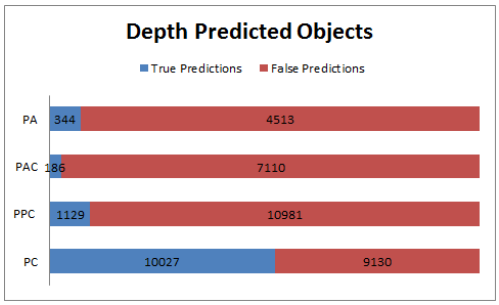
(b) Predicted Objects performance using Incremental Pedestrian Detection for RGB image modality



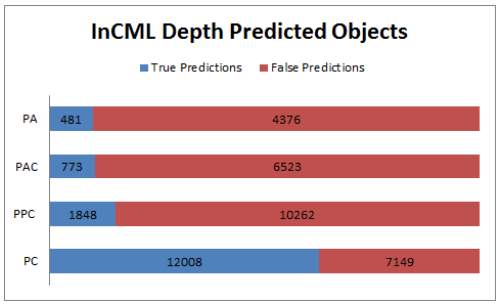
(c) Predicted Objects performance using Uni-Modal Pedestrian for Optical Flow image modality



(d) Predicted Objects performance using Incremental Pedestrian Detection for Optical Flow image modality

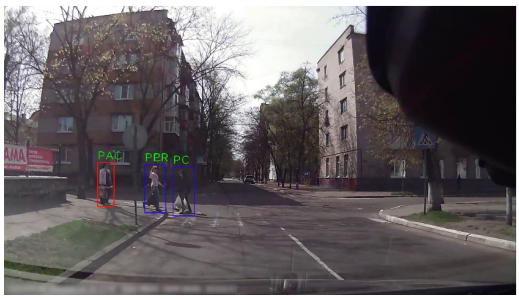


(e) Predicted Objects performance using Uni-Modal Pedestrian for Depth image modality

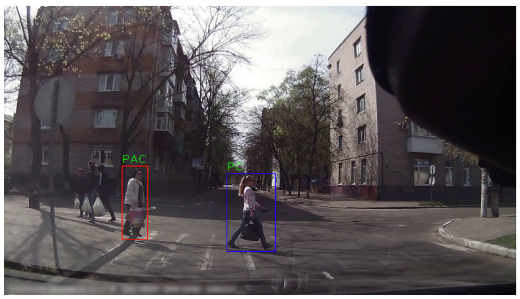


(f) Predicted Objects performance using Incremental Pedestrian Detection for Depth image modality

Figure 2.15 – Comparison of Predicted Objects Actions using Uni-Modal Pedestrian Detection and Incremental Pedestrian Detection. PC= Pedestrian is Preparing to Cross the street, PC= Pedestrian is Crossing the street, PAC=Pedestrians is About to Cross the street and PA= Pedestrian intention is Ambiguous.

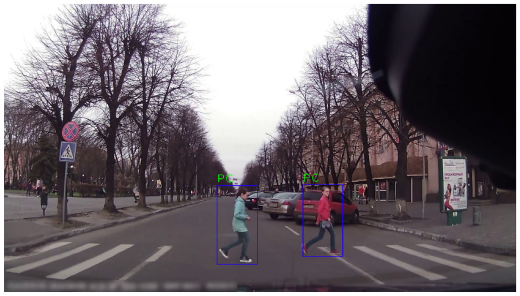


(a) Results of pedestrian detection using PPC, PC, PAC, PA classes.



(b) Results of pedestrian detection using PPC, PC, PAC, PA classes.

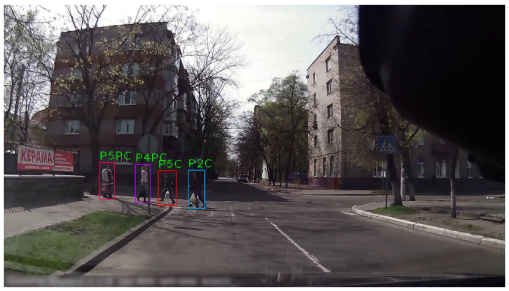
(c) Results of PPC, PC, PAC, PA detection



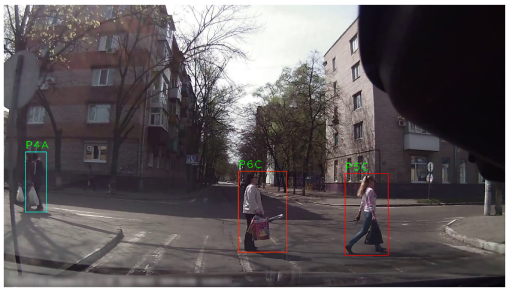
(d) Results of PPC, PC, PAC, PA detection.

(e) Results of PPC, PC, PAC, PA detection

(f) Results of PPC, PC, PAC, PA detection.



(g) Results of pedestrian detection using PPC, PC, PAC, PA classes and pedestrian ids.



(h) Results of pedestrian detection using PPC, PC, PAC, PA classes and pedestrian ids.

Figure 2.16 – Example of pedestrian actions detection using a different approach.

2.6 Conclusion

In this chapter, we systematically studied three methods of integrating different image modalities (RGB, Depth, Optical Flow) to improve the detection component. We used the JAAD dataset, which provides videos only in RGB format. We derived the Optical Flow and Depth image modality in order to apply our Incremental Cross Learning modality.

We analyzed two detection methods:

1. We trained the CNN with all pedestrian samples using the classical detection system where we consider all the annotation tags only as pedestrians;

2. We trained the CNN with all pedestrians using the various action tags;

We studied different pedestrian actions to find out if a pedestrian is crossing the street or if the pedestrian's action presents a critical situation:

1. We split the pedestrian Joint Attention for Autonomous Driving (JAAD) dataset in into four classes: pedestrian is preparing to cross the street (PCC), the pedestrian is crossing the street (PC), pedestrian is about to cross the street (PAC), and pedestrian intention is ambiguous (PA);

2. We extracted the Optical Flow and Depth motion from the JAAD dataset;

3. We trained all pedestrian samples using the pedestrian action tags mentioned above with the RetinaNet using RGB, Optical Flow and Depth motion for pedestrian detection using a Classical-unimodal approach and our Incremental Cross Learning modality;

4. We explored the Classical Uni-modal on RGB image modality using the four proposed pedestrian action tags mentioned above and take into account the pedestrian behaviors and its id according to JAAD pedestrian annotations (e.g., pedestrian1 cross= P1C, pedestrian2 cross=P2C, pedestrian3 ambiguous=P3A).

We evaluated the pedestrian detection approach (called the classical approach), where all samples are tagged as pedestrian and not pedestrian and a pedestrian detection approach using multiple tags. The first method achieved better performance since it has only to distinguish the pedestrians from other road users, in contrast to the second one which even has to recognize pedestrian actions. The second detection approach returned a weaker performance than the classical one. On the other hand, pedestrian detection using the multiple tags approach could be useful for the prediction action part and especially for the Time to Cross (TTC) the street prediction, developed in the next chapter.

The InCML outperformed the classical detection approach on all modalities, but its performance is statistically significant only for the RGB image modality. We noticed that the performance of the InCML detector is directly proportional to the achievements of each pedestrian detection action component.

The InCML approach is based on transfer learning, and the previous CNN directly influences this method. The InCML approach achieves better results due to this transfer learning technique. In the second and third learning steps, the InCML takes more specific and optimal weight information according to the target application. Moreover, this InCML method is a generic model. It is more flexible, allowing

for adaptive settings according to each CNN detector. It could be used and adapted for all types of CNN, not only for a specific one.

In the next chapter we proposed to with CNNs and LSTM is order to predict the pedestrian action intentions and time to cross the street for each pedestrian.

Chapter 3

Pedestrian Action Prediction and Time to Cross Estimation

Contents

3.1 Introduction **75**

3.2 Related Work **78**

3.2.1 Prediction Analysis Models 78

3.2.2 Related Studies Concerning Pedestrian Action Prediction . . 81

3.3 Method **83**

3.3.1 Pedestrian Position and Action Prediction 83

3.3.2 Estimation of Time to Cross 84

3.4 Experiments **86**

3.4.1 Data setup 86

3.4.2 Training protocol 86

3.4.3 Testing protocol 90

3.4.4 Evaluation protocol 91

3.5 Results **92**

3.5.1 Evaluation of Pedestrian Actions Prediction 93

3.5.2 Evaluation of Pedestrian time to cross Component 93

3.6 Conclusion **101**

3.1 Introduction

Pedestrian detection and action/intention prediction is a crucial component of advanced driver assistance systems since it contributes to the road flow safety. The traffic participants security could be significantly improved if these systems could also predict and recognize the pedestrian actions, or even estimate the time to cross the street for each pedestrian.

Human errors abound due to fatigue, driving the car while using the telephone, driving under the influence of medicine, or pedestrians' bad and/or risky behavior any of which may generates traffic collisions. These collisions between cars and pedestrians could be greatly decreased if human error could be eliminated by employing an Advanced Driver Assistance System (ADAS) for pedestrian detection. If these ADAS systems include not only the pedestrian detection but also the pedestrian actions prediction and/or estimate time to cross the street of the pedestrians, the collision could be significantly reduced by adapting the functionality of intelligent vehicles according to the road used concerned. Such a system can dramatically improve the vulnerable road user (e.g pedestrians) safety in future ADAS and/or self-driving car.

In the first chapters, we analyzed the first two components from our main thesis objective: the Perception and the Identification/Fusion. The Perception component involves the stereo vision dataset based on the Daimler dataset [EESG10] while the Identification/Fusion component use the environment information provided from the prior component then detect the pedestrian and classify the pedestrian's action at T=0 (distinguish between pedestrian and non-pedestrian, distinguish between 4 pedestrian's actions: PC,PPC,PA,PAC) based on our Incremental Cross-modality Deep Learning approach. In this chapter we investigate the Decision component (see Figure 3.1) where the module has to estimate the risk in order to identify the appropriate vehicle control level (information/advice). Thus, in this chapter we analyze the pedestrian action prediction and the estimation of pedestrian time to cross the street.

Prediction and estimation are sometimes seem to function similarly, but there is a sharp distinction between them in the standard model of a statistical problem. An estimator uses data to guess at a parameter while a predictor uses the data to guess at some random value that is not part of the dataset. Estimation is the calculated approximation of a result while the prediction is merely assuming something about the future.

Thus, the designed system in this chapter touches the problem of pedestrian action, intention prediction and estimation of time to cross the street of pedestrians recorded from on-board a moving vehicle.

To do that the system should have a high detector component, for localizing and recognize the pedestrians among another road users, a classification component to distinguish the pedestrian actions and a prediction component to estimate the pedestrian actions over next frames (short, medium and/or long time prediction). The prediction component should perform efficiently in different environmental circumstances and should even offer the possibility to estimate the time to cross the street for each pedestrian.

The common approach to solve the road users prediction issue is to involve dynamic factors such as pedestrian trajectory [CS14, EG11], velocity [PESv09], or the expected final goal of pedestrian trajectory [RK15b, KAHS16]. These research inves-

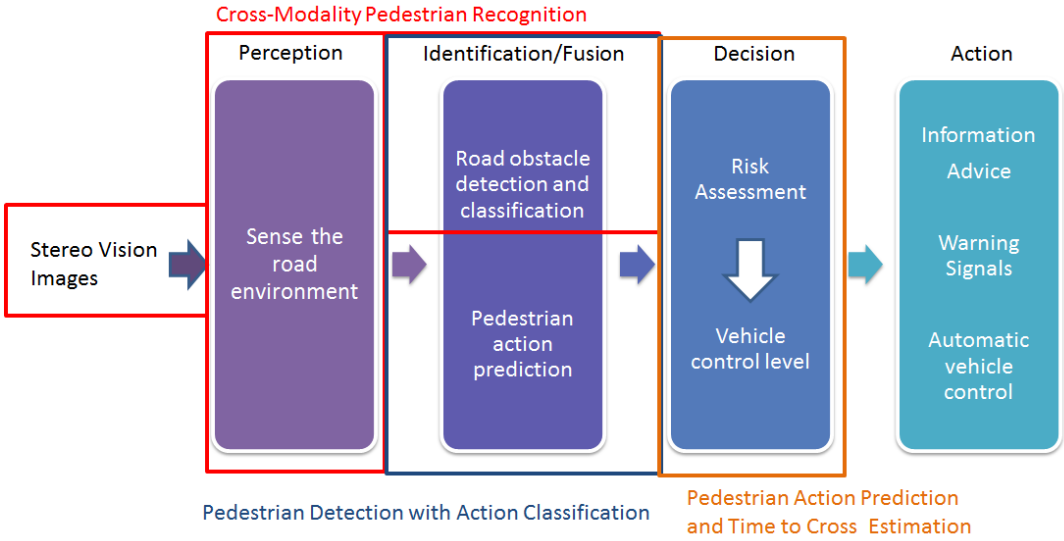


Figure 3.1 – The main architecture of our system. In red are the issues investigated in the first chapter. In blue are the problems studied in second chapter. In orange is the problem analyzed in this chapter

tigations are quite limited in scope because those take into account very few contextual traffic road elements of predicting the behavior of pedestrian. Besides of the spatiotemporal factors which we have mentioned above, there also exist other factors that can influence the crossing behavior of pedestrian, for instance: environmental factors (e.g., weather condition, visibility), the crosswalk structure (e.g., pedestrian traffic light, traffic signs, delineation) or the pedestrian's characteristic (e.g., demographics, the pedestrian's culture).

This issue has been widely investigated, but it still remains an open challenge because progress in pedestrian detection is hindered by the difficulty of detecting all partially occluded pedestrians and the problem of operating efficiently in severe weather conditions. Moreover, current systems cannot yet understand the intention of road users involved to ensure their safety and secure the traffic flow. For this purpose, the system should have i) a detection model for localizing and recognizing the pedestrians among other road users, ii) a classification model to distinguish the pedestrian actions, and iii) a prediction model to estimate the pedestrian actions over the next frames (short, medium and/or long-time prediction). The prediction component should perform efficiently in various environmental circumstances and even offer the possibility of estimating the time to cross the street for each pedestrian.

The difficulty in solving these problem comes from the lack of public annotated data bases. Hence, there are no public databases annotated with pedestrian time to cross, while there are several interesting huge pedestrian detection databases (Kitti, Caltech, among others). The problem is that those databases do not provide any pedestrian action labels. To the best of our knowledge, the only public dataset with pedestrian action tags in urban traffic environmental is JAAD [KRT16]. Since this dataset does not provide the annotations directly for pedestrian time to cross, we determine it for each pedestrian trajectory (frame sequences).

The question is, could we manage the pedestrian action classification and the pedestrian bounding box (BB) detection in one end-to-end detector? or we must use two separate methods: first for pedestrian detection and then for pedestrian

action recognition, as existing approaches from literature?

The contribution of this chapter concerns solving this issue by applying a multi-task deep learning model for detecting, classifying, and estimating the time to cross for multiple pedestrian actors.

The detection and classification components we have already explicitly depicted in the previous chapters, but seldom we have to recall some of the related concepts/approaches previously mentioned the in prior research since the prediction system component is directly connected with the pedestrian detection/classification one.

In this chapter, we focus on pedestrian action prediction and estimation whenever the pedestrian's action presents a risky situation like time to cross the street

To do so, we develop the following methodology relying on a deep learning approach:

- Use the pedestrian action detector based on RetinaNet [LGG+17a] using classical learning approach with RGB image modality. This approach is explicitly presented in the second chapter;
- Use the Joint Attention for Autonomous Driving (JAAD) [KRT16] dataset in our experiments with our four pedestrian action classes which we have described in the second chapter: pedestrian is preparing to cross the street (PPC), the pedestrian is crossing the street (PC), the pedestrian is about to cross the street (PAC), and pedestrian's intention is ambiguous (PA);
- Train a Long Short-Term Memory (LSTM) [HS97] for (T+1, T+2, T+3, T+4, T+5), medium (T+14) and long-time (T+40) prediction in order to estimate the pedestrian bounding box coordinates, the actions and time to cross the street for each pedestrian.

The Chapter is organized as follows: Section 2 outlines some existing approaches from the literature and gives our main contribution. Section 3 presents an overview of our system. Section 4 describes the experiments and the results on the JAAD dataset. Finally, Section 5 presents our conclusions.

3.2 Related Work

Several research activities addressing pedestrian detection have produced significant performances for this issue [BOHS14, ZBO+16, PRNB17, LDWW18, BKFG19].

In terms of data collection, most pedestrian detection systems collect data from video cameras [BOHS14, ZBO+16], from LIDAR [LL16] or fusing data information [SCK16, PRNB17]. These systems are based on handcrafted features models followed by a trainable classifier or deep learning neural networks models. The drawback of these systems is that they cannot anticipate pedestrian actions. The estimation of the pedestrian's intention is even more challenging than the pedestrian detection task because of the pedestrian's ambiguities in the pedestrian's movements. The pedestrian could decide to change his/her behavior/movement in less than one second, an issue which increases the difficulty of solving the problem. Nevertheless, the interest in estimating pedestrian actions for intelligent cars has significantly increased in the last few years [HTDD18, SG13b, RWLS18, RK15a, RRL+18]. In order to find a solution to this issue, the research analyzed has various features like pedestrian movements and/or pedestrian behaviors [FL18, QPLS14], interactions between pedestrians [AGR+16, HJ15b] and pedestrian tracking paths [SG13b, RWLS18].

Most deep learning prediction methods are based on Recurrent Neural Networks (RNNs) because the recurrent connections of RNN allow memorizing historical information from previous states, which is very different from the classical neural network. This ability permits the RNN network to detect changes over time.

The pedestrian action prediction and time to cross estimation can be considered as a sequence prediction problems. Therefore, we briefly review the main Recurrent Neural Networks (RNNs) which could be used in the prediction task followed by some related pedestrian prediction work based on deep learning methods.

3.2.1 Prediction Analysis Models

We briefly present some variation of the RNN which were used in the prediction task in the following:

- **Recurrent Neural Network (RNN)** is based on the hypothesis that all inputs and outputs are self-supporting. The RNN can be seen as multiple copies of the same network where each network transfer the information to its successor (see Figure 3.2). It has minimum a hidden layer at a time which depends on the input at time t,x_1 but also on the same hidden layer at time t-1 or on the output at time t-1. Therefore the RNN has a loop from the hidden layer to itself or from the output to the hidden layer;

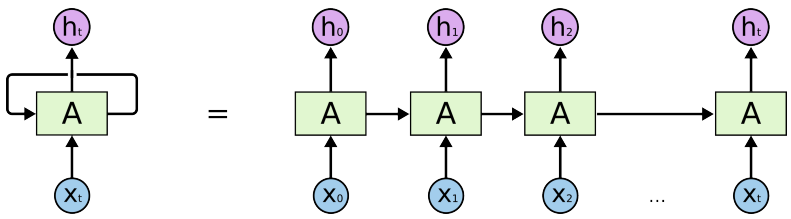


Figure 3.2 – The RNN unrolled architecture

- **Long Short-term Memory (LSTM)** [HS97] is a variation of the recurrent neural network. It was introduced to to learn long time dependencies. An LSTM cell contains at time t, a state C_1, and an output h_t. As input, this cell at time t contains x_t, C_{t-1}. An LSTM has three gates, to enable or not send the information i.e., the forget gate, input gate, and output gate (see Figure 3.3);

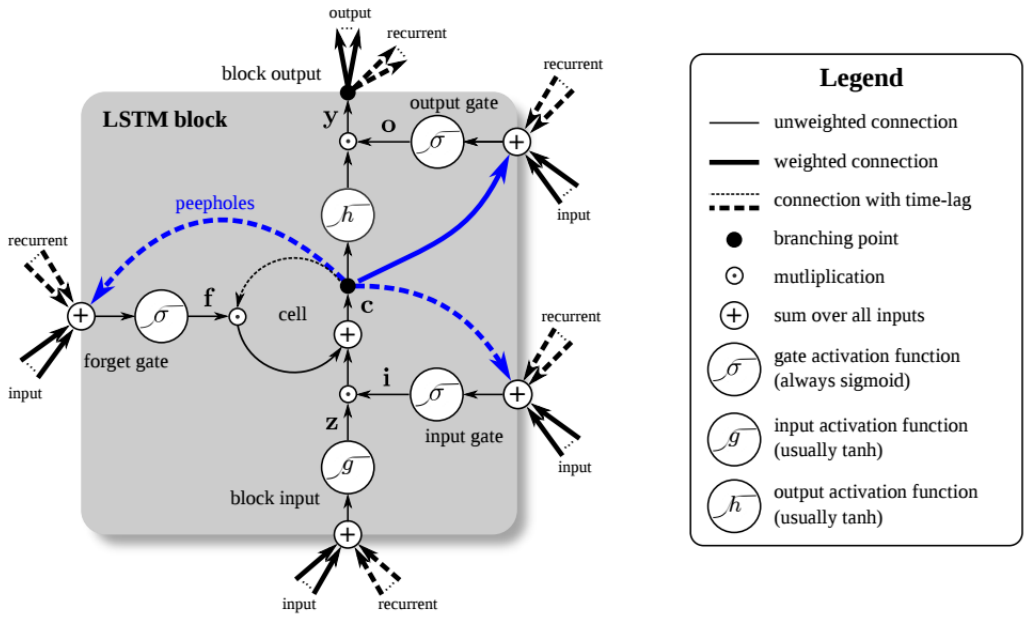


Figure 3.3 – The LSTM representation

- **Gated Recurrent Unit (GRU)**, proposed by [CvMG+14]. It is a variation of the LTSM model. It has fewer parameters than LSTM because it joins the forget and input gates into a single update gate. The GRUs returns better performance than LSTM on particular smaller data sets and same performance on certain tasks. It joins the forget and input gates into a single update gate (see Figure 3.4);

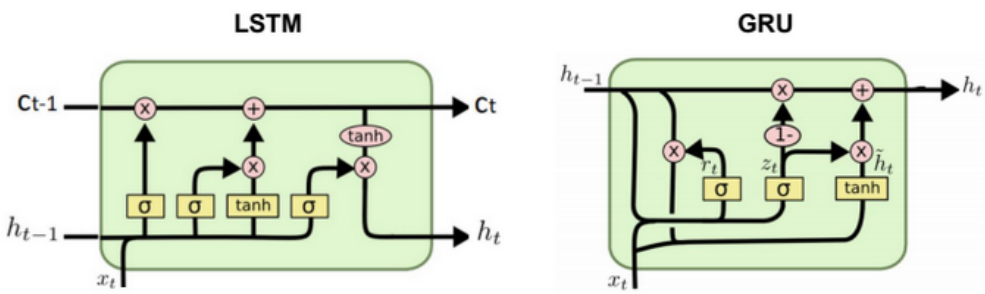


Figure 3.4 – The LSTM vs GRU architectures

- **Recursive Neural Network (RvNN)** proposed in [IC14], is a variety of deep neural network created by implementing the corresponding set of weights recursively over a structured input. In the common RVNN structure, the nodes are using a weight matrix root to share over the entire network (see Figure 3.5);

- **Sequential CNN proposed** in [JZ15,GAG+17], differs from regular works which use RNN to encode the time series inputs. Sequence to sequence learning

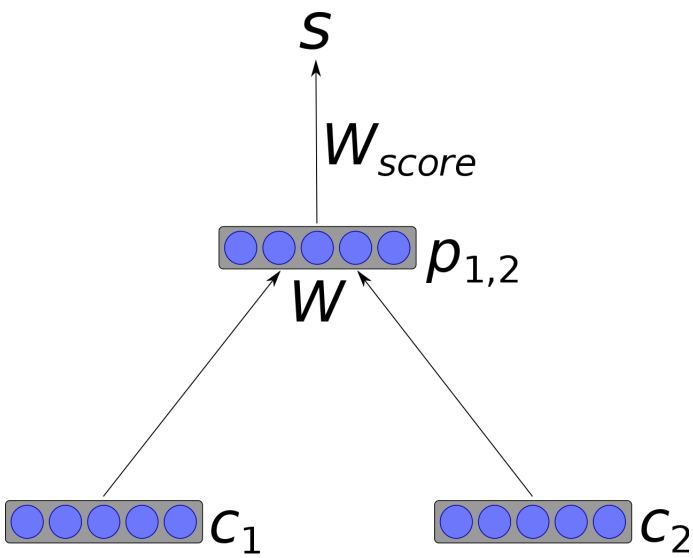


Figure 3.5 – The RvNN architectures

approach implies the same idea of recurrent neural network-based encoder-decoder architectures (see Figure 3.6). The encoder RNN has a sequence x of m elements an input and returns state descriptions z. The decoder RNN uses z descriptions and forms the output sequence y. In order to get the output y $_{i+1}$, the decoder calculates a new hidden state h_{i+1} relying on the prior state h_i, an embedding g_i of the preceding target y_i, as well as a conditional input c_i obtained from the encoder output z. Based on this standard pattern, several encoder-decoder architectures have been proposed, which differ mainly in the limited input and the type of RNN. For instance, in [JZ15] the authors use a CNN instead of on RNN to compute the intermediate between encoder and decoder states.

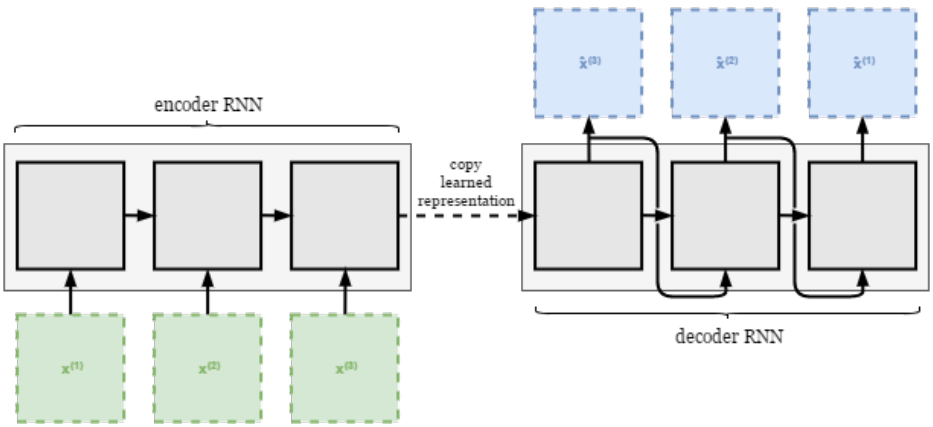


Figure 3.6 – The RNN based encoder-decoder architectures.

Since the Long Short-Term Memory (LSTM) network is the perhaps most successful RNN, and we found it used in many pedestrian prediction approaches from literature, we have decided to use it in our research.

3.2.2 Related Studies Concerning Pedestrian Action Prediction

A comprehensive review of the predicting pedestrian behavior research is presented in [RRL+18] which includes several pedestrian action and movement estimation approaches and also sets out the advantages and shortcomings of the currently available datasets. The authors assume that in the prediction of pedestrian intention it is better to use pedestrian specific dynamics information and the contextual scene.

In [RKT17a], the authors present a pedestrian actions prediction approach based on AlexNet handling JAAD dataset, where they investigate whether the full pedestrian body and part of pedestrian body (consisting either of the head or lower pedestrian body) influence the classification task. They also use a linear SVM to distinguish the situation of a pedestrian crossing or not based on pedestrian attention information. The authors conclude that it is better to use the contextual information to increase the prediction performance. From our point of view, the authors did only a pedestrian action recognition because their approach can only predict the pedestrian action for one-time step ahead (T+1), which we deem, it is not that much of a prediction model..

A pedestrian position estimation based on the Extended Kalman Filter (EKF) and Interacting Multiple Model (IMM) algorithm using Constant Velocity (CV), Constant Acceleration (CA) and Constant Turn (CT) is proposed in [SG13b]. The authors also introduce a dataset, the Daimler dataset, with four pedestrian actions called: crossing, bending in, bending out, and stopping. A combination of the Gaussian process dynamical models, Probabilistic Hierarchical Trajectory Machine (PHTM) and, Kalman Filter and Interacting Multiple Model based on the Daimler dataset using stereo vision images is presented in [KG14]. The authors get better performance than their previous work [SG13b] for the stopping situation. They also make a comparison between these approaches and conclude that the performances almost similar.

A short-term prediction of pedestrian behaviors using Daimler datasets was included in [HTDD18] which is based on the Variational Recurrent Neural Network, which provides the latent variables suitable for a dynamic state-space model. The authors predict whether a pedestrian is stopping or crossing, and obtain high performance on the Daimler benchmark. To predict the pedestrian trajectory and its final destination, an approach using CNN, LSTM and path planning is presented in [RWLS18]. This system can predict both destinations and pedestrian trajectories. A mixture of CNN-based pedestrian detection, tracking and pose estimation to predict if the pedestrian cross the street based on the JAAD dataset is addressed in [FL18]. The authors utilize the Faster R-CNN object detector based on VGG16 CNN architecture for the classification task, use a multi-object tracking algorithm based on the Kalman filter, apply the pose estimation pattern on the bounding box predicted by the tracking system and finally use the SVM/Random Forest to classify the pedestrian actions (Crossing /Not Crossing).

All these approaches for pedestrian action prediction exploit a standard pedestrian detection component which only discriminates between the pedestrian from non-pedestrian, and estimate the pedestrian action or its final destination for the next frames (short, medium and long term).

One of the main goals of our thesis is to predict the future action and location (as bounding box coordinates) of pedestrians filmed from a moving vehicle. We study the short (from T+1 up to T+5), medium (T+14), and long (T+40) time prediction in order to predict the pedestrian's next actions on the JAAD [KRT16] dataset.

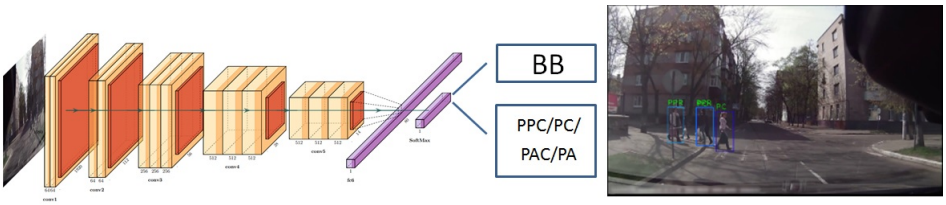


Figure 3.7 – Pedestrian detection using multiple tags

Since the pedestrian prediction component is directly connected with the pedestrian detection one, our pedestrian action prediction can be separated into two stages:

1. We train the RetinaNet [LGG+17a] object detection network to output the four labels, the pedestrian is preparing to cross the street (PPC), the pedestrian is crossing the street (PC), the pedestrian is about to cross the street (PAC), the pedestrian's intention is ambiguous (PA) along with the bounding box, instead of outputting just the "Pedestrian" label (see Figure 3.7). This method is described in second chapter;

2. We fuse the output (the bounding box coordinates) of the RetinaNet detection network with an LSTM, in order to estimate the pedestrian action intentions and its location.

Several advantages of LSTM for prediction, in comparison with MLP or Kalman filter, among others, are well known in theory and literature, fact that we decided to use LSTM in our model. Thus, the recurrent connections of RNN allow to memorize historical information from previous image frames, while MLP only uses information from the actual frame. This ability permits the RNN network to detect changes over time. Hence, a comparison between Kalman Filter and RNNs for signal estimation was made in [DH92], It shown that RNNs improve signal estimation compared to Kalman filter.

We also present a multi-task application which can estimate not only the time to cross for each pedestrian but also its actions.

The time to cross estimation task is more challenging and difficult than estimating the next pedestrian action due to the lack of public annotated data. This task, is sometimes challenging even for a human being because pedestrian movements are unpredictable. To our knowledge, there are no different approaches for pedestrian time to cross (TTC) prediction, other than the method addressed in [FL18] on JAAD dataset. Nevertheless, the authors in [FL18] have handled this problem in a step-by-step manner, including the pedestrian tracking component, based on different image processing and machine learning approaches, allowing finally for the pedestrian TTC prediction. We propose an original method for TTC prediction, without an explicit tracking component, based only on deep learning neural networks.

The JAAD dataset is not annotated for the prediction of pedestrian time to cross issue. The issue of TTC prediction is addressed in [FL18] where the authors made their own pedestrian TTC annotation on JAAD dataset to solve it, but the authors did not make public these annotations. Moreover, the authors did not apply their annotation process on all JAAD videos, but only on several sequences. For the TTC prediction problem, we select some cues from the JAAD [KRT16] public dataset in

order to solve this issue and then we made our pedestrian TTC annotation for all videos.

We also present a multi-task application which can estimate the time to cross the street for each pedestrian using a recurrent neural network approaches (LSTM) in two ways:

- using only BB coordinates in order to estimate the time to cross the street (see Figure 3.8);
- using BB coordinates and pedestrian action tags in order to estimate the time to cross the street (see Figure 3.9);

We use the classical approach where the detection and prediction part were independently analyzed (we called the two-stage approach). The LSTM estimate the time to cross street for each video sequence (estimate the time to cross for all pedestrians from the entire visual spectrum).

3.3 Method

In this section, we outline the components and methods used for solving this issue. Since the pedestrian detection component was depicted in the second chapter, in this section, we explicit describe the pedestrian action prediction component and the pedestrian time to cross estimation one.

3.3.1 Pedestrian Position and Action Prediction

The conventional approach for solving the difficulty of pedestrian behavior prediction is to employ a minimum of one dynamic elements contributing to the perception of pedestrian behavior situations such as trajectory [HTDD18], or velocity [SG13b], or to anticipate the final destination of pedestrians [RWLS18].

Some researches investigate the effectiveness of pedestrian's contour, body language and posture to predict their intention.

Moreover, to achieve a high pedestrian action and movement prediction performance, it is necessary to take into account the temporal context information in order to help predicting the pedestrian behavior.

The prediction issue is commonly grouped into two categories:

1. Collision avoidance scenarios (short-term modelling), where the goal is to react with emergency maneuvers for road obstacle avoidance. The prediction horizon is here max. 1-2 seconds [RK15a, RRL+18].
2. Long-term modelling, where the goal is to have a more comfortable driving behavior. The prediction horizon here is 2+ seconds, depending on the vehicle speed and ruttier environment [RK15a].

We focus on the short (from T+1 up to T+5), medium (T+14) and long (T+40) term prediction approaches of both pedestrian position and action by using an LSTM to take into account the temporal context information (previous frames from T-5, T-14 and T-40). The LSTM input are 2D bounding box (BB) coordinates provided by the detection component mentioned above.

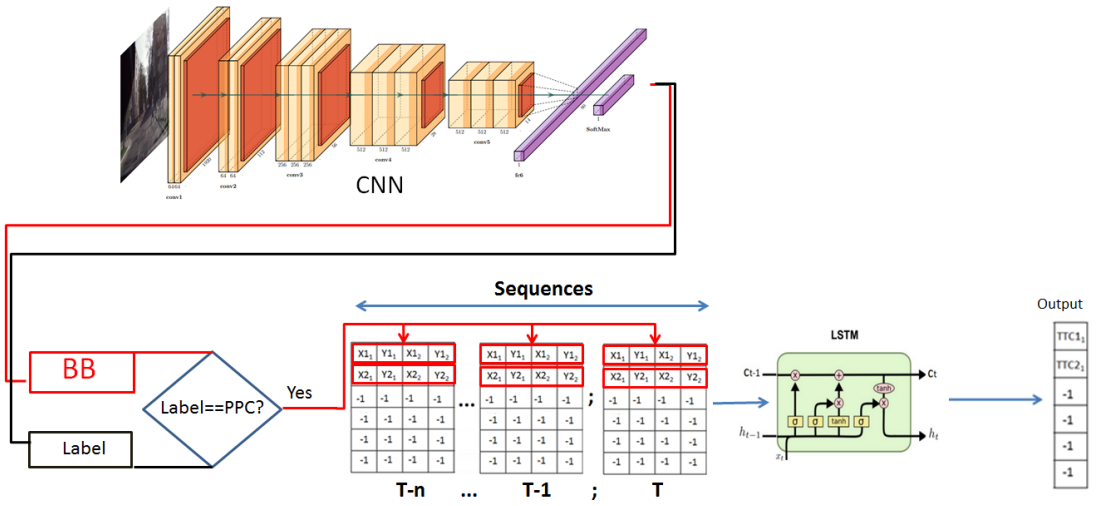


Figure 3.8 – Our time to cross the street estimation method using only BB coordinates in order to estimate the time to cross the street. BB= Bonding Box coordinates, Label= Pedestrian action tag, PPC= Pedestrian Preparing to cross the street; TTC= time to cross; -1= no pedestrian.

Whenever applying the pedestrian detection method, the LSTM input data are the pedestrian tags (label class) and BB coordinates which allow to anticipate the next frames following the pedestrian BB coordinates and his/her behavior (see Fig 3.10).

Our prediction model consist in four blocks of LSTM with 50 nodes followed by Dropout layer with a rate of 20% for each LTMS layer and finally a two fully connected layers with four and respectively one neurons. We used this architecture because we observed a better performance on these values. These values have been tuned over a validation dataset.

3.3.2 Estimation of Time to Cross

The estimation of time to cross for each pedestrian is essential for the ADAS systems since it could predict if and when there could be a risky situation.

From a machine learning point of view, TTC estimation can be considered as a regression problem, where we aim at estimating an integer or a real value (whether we consider a number of frames or a time in seconds) for each frame of a given video. As the dynamic of the signal is essential to estimate TTC efficiently, we have naturally turned toward the use of a recurrent neural network to capture the spatial, temporal context of the motion. Among recurrent models, we have chosen to use LSTMs which have shown their efficiency on many sequence analysis problems.

To predict the pedestrian time to cross, we proposed two approaches:

- individual estimate of TTC for each pedestrian BB sequences provided by the pedestrian detector (using only PPC samples);
- multiple estimates for all detected pedestrians (using all samples).

The prediction model is based on LSTM, and it has the 2D bounding box (BB) coordinates as input data provided by the detection component. The output consists of time to cross for each pedestrian, and it outlines over how many frames the

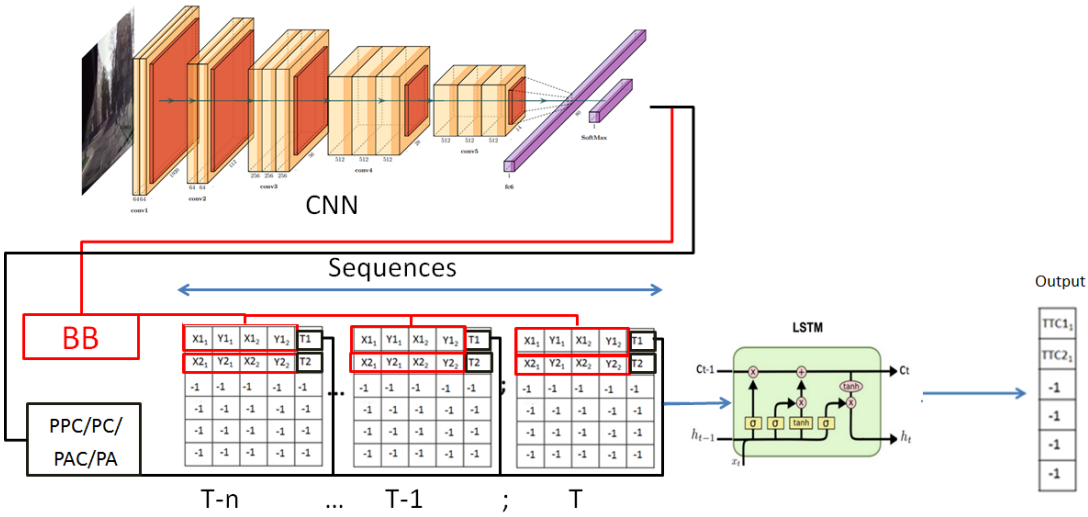


Figure 3.9 – Our time to cross the street estimation method using the BB coordinates and pedestrian action labels in order to estimate the time to cross the street. BB= Bonding Box coordinates, PPC= Pedestrian is Preparing to Cross the street; PC= Pedestrian is crossing the street; PAC= Pedestrian is About to Cross the street; PA= Pedestrian's intention is Ambiguous; P1,P2= Detected pedestrians; T1,T2= Pedestrian Action Tags; TTC= time to cross; -1= no pedestrian.

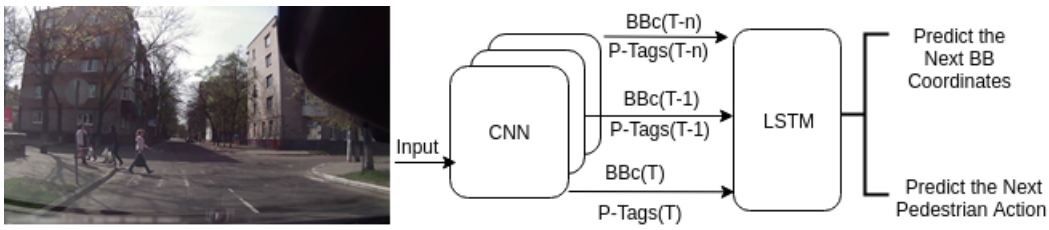


Figure 3.10 – A unified CNN-LSTM architecture for detection, recognition and pedestrian action prediction. BBc=Bounding Box Coordinates, P-Tag=Pedestrian action tag, T=Time step, n∈ {1,14}. The CNN has the frames as input data and the LSTM has the pedestrian BBc and pedestrian action tags as input data.

pedestrian crosses the road. We take into account the temporal context information for the previous frames from T-5, T-14, and T-40.

We emphasis that the detection and prediction components are learnt independently.

The detection step is based on RetinaNet [LGG+17a],because its performance exceeds the Faster R-CNN [RHGS15a], R-FCN [DLHS16], SSD [LAE+15] and YOLOv1 [RDGF15]. It has as input the entire RGB images and returns the pedestrian corresponding bounding box and its action tag.

The prediction model is based on LSTM, and it has the 2D bounding box (BB) coordinates as input data provided by the detection component. The output consists of time to cross for each pedestrian, and it outlines over how many frames the pedestrian crosses the road. We take into account the temporal context information for the previous frames from T-5, T-14, and T-40 in order the estimate the time to cross the street in term if short (5 frames), medium (14 frames) and long (40 frames) term estimation.

3.4 Experiments

In this section, we present our set of experiments, including setups and performance assessment of our approaches.

3.4.1 Data setup

There are many different large-scale pedestrian detection datasets used for the pedestrian detection task. Several of them have in somehow the potential of derivate specific annotation to estimate/predict the pedestrian behaviors/actions. Most of the data sets provide only the bounding boxes annotations, from which is quite hard to make your annotations in order to solve the prediction of the pedestrian action and estimate the pedestrian's time to cross. Hence there is a lack of public annotated dataset, the fact that the pedestrian action prediction and time to cross estimation is more charging and difficulty to solve.

To our knowledge, there are only two data sets which could be used in order to solve these issues. One of them is Daimler [SG13a] which provides four different pedestrian motion types crossing, stopping, starting to walk and bending-in, the bounding box coordinates, and the trajectory data. The drawback of the Daimler dataset is that it provides the video sequences with only one pedestrian in the series and the pedestrians are not occluded.

We did not use the Daimler [SG13a] dataset, because it is too small to train our deep learning model and it was not acquired in real urban traffic conditions and shows a single pedestrian per video performing predefined actions.

The only public dataset with pedestrian action tags in urban traffic environmental is JAAD [KRT16].

This dataset provides pedestrian bounding boxes (BB) for pedestrian detection (including for several of them the pedestrian actions), pedestrian attributes for estimating the pedestrian behavior and traffic scene elements.

The drawback of JAAD dataset is that it does not provide annotation for the pedestrian time to cross estimation task. This issue struggles us to make our pedestrian TTC annotation for all videos. Nevertheless, the only public dataset with pedestrian action tags in urban traffic environmental is JAAD [KRT16] fact that we decided to use in our experiments.

3.4.2 Training protocol

The training protocol is related to the pedestrian detection component one since the prediction involves the detection as a prior stage. We briefly outline the common training settings, and we explicitly detail for the prediction step. Thus, we used the first 250 video sequence for training process and the rest for the testing because the videos are disjointed. The training and testing samples include even the partially occluded and heavily occluded BBs.

In [RKT17b, RKT18], the authors present a variety of pedestrian behaviors done before crossing and after crossing the street and even when the pedestrian does not cross the street. These behaviors were collected and annotated with different action labels according to the pedestrian crossing attributes (we called crossing event) for each pedestrian from all video sequences. We use the event terms instead of cross-

ing attributes to make a clear distinction between pedestrian action, pedestrian behavior and crossing attribute.

The event reprints the main action of the pedestrian that happens on the video. The pedestrian action represents the pedestrian act that occurs in a specific place during a particular interval of time. The pedestrian behavior represents the manner and/or reaction of the pedestrian before, during and after given urban traffic circumstances.

The events could be:

- the pedestrian completes to cross the street;
- the pedestrian has no intention to cross the road (e.g. sits on a public bench, waiting for public transportation);
- the pedestrian does not cross the street (e.g the pedestrian has started to cross the street but suddenly he/she is stopping).

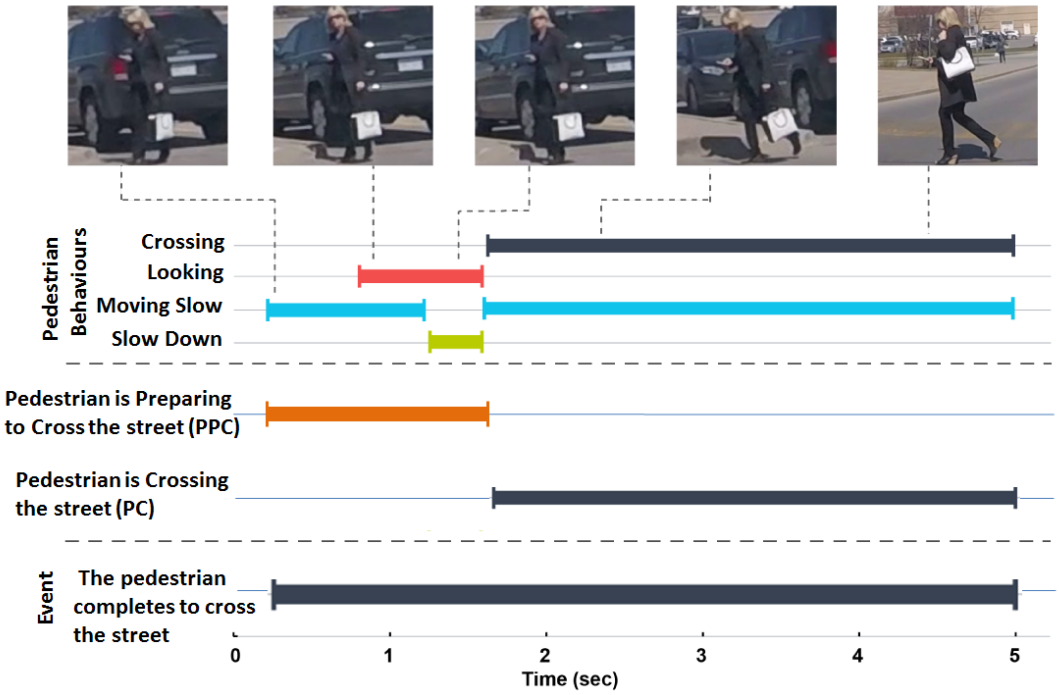


Figure 3.11 – Timeline of event, action and pedestrian behavior whenever the pedestrian is going to cross the street. This image was picked from JAAD [KRT16] dataset source and modified/updated to our requirements.

For instance, if the pedestrian is going to cross the street, he/she can present minimum behavior like standing, looking, and then crossing the street, or moving, looking, and then crossing the street. The pedestrian actions applied before or during one event, could be different for each pedestrian, even if the event is the same. Hence, according to these action annotations, we can observe that there exists a typical pattern for each pedestrian for a given event (see Figure 3.11).

We observed that the JAAD event annotations do not adequately cover all the pedestrian intention, and it could not directly help us to solve the pedestrian action prediction and time to cross estimation.

We wondered what happens if the pedestrian is walking along the street (there is no footpath/crosswalk) or the pedestrian already passed the road, and maybe he/she suddenly decide to return. We wondered if could we make a clear distinction between the pedestrians who are planning to cross the street and then cross the road and the pedestrian who starts to cross the street but for a different reason he/she stops.

These particularly cases struggle us to derivate our annotation in order to fulfill our thesis objectives: pedestrian action prediction and time to cross estimation.

Therefore, according to the specifications and annotations presented above, we separate the pedestrian labels into four classes. In the second chapter, we have detailed the explication of the pedestrian labels, but we sketch them in this chapter for a better understanding of our approach:

1. Pedestrian is Preparing to Cross the street (PPC), where the pedestrian is walking or standing, pays attention or not and changes or does not change its behavior before crossing. In this case, the pedestrian is definitely assume to cross the street after these actions.

2. Pedestrian is Crossing the street (PC), where the pedestrian is observed from the point of crossing until he/she has crossed the road. There are video sequences where the pedestrians are annotated only from the point of crossing the street.

3. Pedestrian is About to Cross the street (PAC), where the pedestrian is about to cross and pays attention and responds according to the event. The pedestrian definitely does not cross the street after these actions.

4. Pedestrian intention is Ambiguous (PA), where the pedestrian is walking or standing, and his/her intention is ambiguous. In this case, the pedestrian has crossed the road or other event which does not present a risk situation.

The Detection Learning Protocol including all learning setups are detailed in the second chapter. In this chapter we used the RetinaNet [LGG+17a] as a pedestrian detection algorithm using RGB image modality.

Pedestrian Action Prediction Setups

The Long Short-Term Memory (LSTM) use as input the bounding box (BB) coordinates and pedestrian action tags in order to predict the pedestrian action for the next frames as output: short (T+1,T+2,T+3,T+4,T+5), medium (T+14) and long (T+40) term.

Each video sequence has a different number of frames and pedestrians per frame. To create the training set and to ensure the pedestrian information is not mixed with those of others pedestrians information, we create a generalization method to track and provide the data for each pedestrian. This method tracks each pedestrian on each video sequence from the first point of performing in a video until the pedestrian disposes of the frames and creates a subset with all bounding box (BB) coordinates and pedestrian actions for each pedestrian. For instance, the video no 1 has 600 frames with two pedestrians which have annotations for their actions. Both pedestrians appear from scarce, but after several frames, one of the pedestrians disappears from the ruttier environment. Whenever one pedestrian exits from

the environment or the video sequence is over, the method creates an independent subset with the according annotations for each pedestrian. Those subsets generate the training dataset, which consist of various pedestrian independent sequences with different lengths.

In our approach, the training set has the bounding box (BB) coordinates and the pedestrian action as data information in contrast with the usual methods from literature which use the centroid of the bounding box (x,y) coordinates.

We performed the LSTM training process with the ADAM learning algorithm method, with ten epochs using pasted time steps of 5, 14 and respectively 40 frames in order to predict the next pedestrian actions on the next frames (T+1, T+2, T+3, T+4, T+5, T+14, and T+40).

Estimation of time to cross Protocol

The pedestrian time to cross was calculated only for pedestrians who are preparing to cross the street (PPC) because only in this particular case are the pedestrians definitively going to cross the street and only in this specific case can we estimate the time to cross for each pedestrian. Thus after a PA action, the pedestrian will never cross the street, and after a PAC, the crossing is quite unpredictable (even for the pedestrian itself).

To determine the time to cross, we use an LSTM which is trained independently of the CNN based detector since it is applied after the detection step.

We create a bounding box matrix to predict the time to cross for multiple pedestrians sequences within the LSTM (see Figure 3.12).

The LSTM was learnt with the following methodology:

- We created an input bounding box matrix (4x20) for each frame where we set the bounding box coordinates only for the pedestrians preparing to cross the street (see Figure 3.8). For the other pedestrians (PA, PC, PAC) the input values in that matrix are fixed to (-1) indicating there is not any pedestrian preparing to cross the street. For the PPC element in the input matrix, the corresponding output is the time to cross, which consists in the descending scrambling order of frames to the moment of crossing. While for the other pedestrians (PA, PAC, PC), the corresponding output is (-1). In our approach, we consider there are no more than 20 pedestrians per frame.

- We created an input bounding box matrix (5x20) for each frame where we set the bounding box coordinates and pedestrian action tag only for the pedestrians preparing to cross the street. The input values for the other pedestrians action tags (PA, PC, PAC) are coded identically as previously matrix box while for PPC is coded with 0. For the PPC element in the input matrix, the corresponding output is the time to cross, which consists in the descending scrambling order of frames to the moment of crossing, while for the other pedestrians (PA, PAC, PC), the corresponding output is (-1).

- We created an input bounding box matrix (4x20) for each frame where we set only the bounding box coordinates for all the pedestrians actions. The output matrix is the time to cross for the PPC tag, while for the other pedestrians (PA, PAC, PC), the corresponding output is (-1).

- We created an input bounding box matrix (5x20) for each frame where we set the bounding box coordinates and pedestrian action tag for all the pedestrians actions (see Figure 3.9). The input matrix values for the pedestrians action tags are coded with the following: PPC=0; PA=1; PC=2; PAC=3. The output matrix is the time to cross for the PPC tag, while for the other pedestrians (PA, PAC, PC) the corresponding output is (-1).

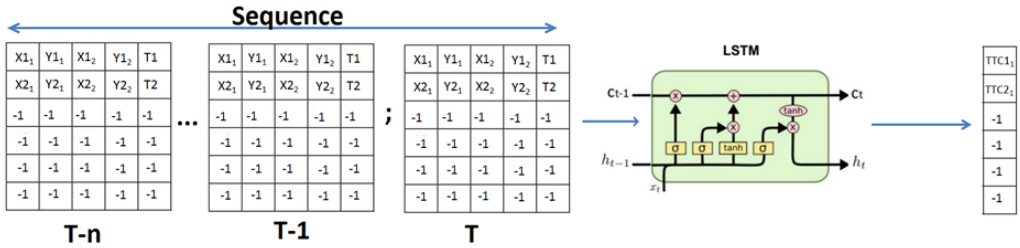


Figure 3.12 – The proposed LSTM based architecture for pedestrian time to cross estimation. Input: The BB matrix (4 x 20) at frame T until previews T-n (n= 5, 14, 40), where the X_{i1},Y_{i1},X_{i2},Y_{i2} i=1 to 20 are the BB coordinates for each pedestrian i detected on frame T; output: TTC(i)= number of frames from frame T to the beginning of crossing for the pedestrian i; -1= no pedestrian.

We performed the LSTM training process with the ADAM learning algorithm method, using previous time steps of 5, 14, and respectively 40 frames to estimate time to cross. For each step, the LSTM estimates over how many frames the PPC pedestrian will cross the street.

3.4.3 Testing protocol

The testing set used to assess the CNN, and LSTM model performances are independent of the training dataset. It contains 105 video sequences. It has a total of 43420 samples, where 12110 examples are pedestrians who are preparing to cross the street (PPC), 19157 samples are pedestrians who are crossing the street (PC), 1296 samples are pedestrians who are about to cross the street (PAC) and 4857 examples where intention is ambiguous (PA). We test the prediction part on the difference frames to analyze the performance of our prediction model. The TTC model was assessed only on the 12110 samples, of pedestrians who are preparing to cross the street (PPC) because this is the only case where the pedestrians are clearly going to cross the street.

We test the prediction component on two different ways:

- first only on the 12110 pedestrian samples to assess only the predictor capabilities independently of the pedestrian detector and classifier, because this is the only case where the pedestrians are clearly going to cross the street.

- second on the all 43420 pedestrian samples.

The TTC is tested on the ground truth (real values) test BB samples which are provided by JAAD dataset and also on the detected BB samples which are supplied

by the pedestrian detection component, while the pedestrian action prediction is tested only on the grand truth values.

Our testing methodology consists of the upcoming plan:

- testing the pedestrian position prediction and the action prediction component independently of the previous detection component;
- testing, independently of the detection and classification components, the pedestrian time to cross estimation on the PPC pedestrian samples only with and without pedestrian action tags;
- testing, independently of the detection and classification components, the pedestrian time to cross estimation on all pedestrian samples with and without pedestrian action tags;
- testing the detection component connected with the prediction component (time to cross).

In this section, we perform all the testing steps.

3.4.4 Evaluation protocol

The evaluation process for all the CNN models was done with the Tensorflow Deep Neural Network Framework. The performances were assessed by the Accuracy (ACC) and Root Mean Square Error (RMSE) for the prediction component. The ACC values were computed using the Keras metrics tool, where P represent the predicted value (observed value), A the Actual value (true value) and n the number of samples.

$$\mathrm{ACC} = \frac{1}{n}\sum_{i=1}^{n} 1(\mathrm{P}_i - \mathrm{A}_i). \tag{3.1}$$

Moreover, we compute Root Mean Square Error (RMSE) using the Scikit-Learn tool [PVG+11], in order to measure the differences between the predicted values and the observed ones, which is the common estimator evaluation metric (deviation of the prediction errors).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\mathrm{P}_i - \mathrm{A}_i)^2}{n}}. \tag{3.2}$$

In this equation, P represents the Predicted value (expected values), A the Actual values (true value, actual results) and n represents the number of sequences.

For TTC we calculate the Mean of Root Mean Square Error (MRMSE) for each video, because of the huge number of pedestrians per video.

$$\mathrm{MRMSE} = \frac{1}{m}\sum_{j=1}^{m}\sqrt{\frac{\sum_{i=1}^{n}(\mathrm{P}_i - \mathrm{A}_i)^2}{n}}. \tag{3.3}$$

In this equation, P represents the Predicted value (expected value), A the Actual values (true value), n represents the number of frames/video and m the number of videos.

We calculate the margin of error (Confidence Interval - CI) to evaluate whether one model is statistically better than another one.

$$\mathrm{CI} = 1.96\sqrt{\frac{\mathrm{P}(100-\mathrm{P})}{n}}\%. \tag{3.4}$$

In this formulation, P represents the performance system (e.g., ACC, RMSE) and n represents the number of testing samples.

3.5 Results

The experiments were performed on the JAAD dataset using the original video size. We independently provide the results for the pedestrian, action precision results and finally, we present the estimation of time to cross methods. The performance of the pedestrian detection component are presented in second chapter.

Table 3.1 – The performance of the pedestrian action prediction. ACC BB: Accuracy estimation for the the next bounding box coordinates; RMSE BB: The bounding box Root Mean Square Error; ACC Actions: The actions accuracy estimation for the the next frames; RMSE Actions: The Root Mean Square Error action prediction; ACC Model: The accuracy of the model which is a mean between the ACC BB and ACC Actions; RMSE Model: The Root Mean Square Error is a mean between the RMSE BB and RMSE Actions

Past Time Steps	Next Frames	ACC BB	RMSE BB	ACC Actions	RMSE Actions	ACC Model	RMSE Model
5	1	73.45	0.092	97.23	0.077	85.34	0.0845
	2	72.54	0.1096	97.40	0.127	84.97	0.1185
	3	71.66	0.1230	97.12	0.1637	84.39	0.1434
	4	71.67	0.1469	92.27	0.2226	81.97	0.1840
	5	69.47	0.1739	84.74	0.2726	77.11	0.2233
	14	58.46	0.2171	84.68	0.3172	71.57	0.2672
	40	31.94	0.3159	78.96	0.3623	55.45	0.339
14	1	72.51	0.0866	97.57	0.0601	85.04	0.0733
	2	71.20	0.0988	97.27	0.0986	84.23	0.0987
	3	70.73	0.1091	96.97	0.0951	83.85	0.1021
	4	68.79	0.1319	96.62	0.1093	82.71	0.1206
	5	66.53	0.1422	96.35	0.1169	81.44	0.1295
	14	60.24	0.2114	91.65	0.2206	75.94	0.2160
	40	24.00	0.3560	78.16	0.3982	51.08	0.3771
40	1	76.13	0.0800	97.32	0.0563	86.72	0.0681
	2	69.43	0.1219	96.99	0.0782	83.21	0.1000
	3	68.73	0.1227	96.69	0.1026	82.71	0.1122
	4	69.91	0.1330	96.40	0.1121	82.65	0.1225
	5	67.09	0.1425	95.91	0.1430	81.86	0.1449
	14	60.24	0.235	92.53	0.2260	76.38	0.2306
	40	32.85	0.2866	82.31	0.2914	57.58	0.2890

3.5.1 Evaluation of Pedestrian Actions Prediction

We present a comparison between our prediction models in Table 3.1. We also present the pedestrian prediction action on different time steps. According to RMNS Model values, we observe that our model achieved the best performance using 40 frames as a prior time step to predict the next 1, 3, 4, and 40 frames. The values obtained the best performance using 14 frames as a prior time step to predict the next 2, 5 14, frames. The RMNS Model values obtained using 5 frames as previous time steps returned the worst performance.

The smallest RMSE error is the best one, but the slight differences between RMSE are not relevant. We observed that with the increase in prediction time, the action estimation is more unpredictable, but by using more time steps to predict the next frames the prediction becomes more stable and accurate. The low performance using more previous time steps to predict the next frames can come even from the pedestrian behaviors manifested during stable time, where the pedestrian's actions could often be shifting, a fact which affects the temporal information. The pedestrian prediction process is a complex process for smart systems, but it is also often difficult for human beings, since the pedestrians can change their behavior suddenly.

3.5.2 Evaluation of Pedestrian time to cross Component

Table 3.2 – The estimation of time to cross method, independently of the detection-classification component. PPC:Pedestrian is Preparing to Cross the street. Real values: testing, independently the pedestrian time to cross estimation on the all real pedestrian samples; Detected Values: testing the detection component connected with the prediction component (time to cross).

Learned on		Tested On	Past Time Steps	MRMSE	
				Real Values	Detected Values
Only PPC BB Coordinates	With Action Tag	All Samples	5	12.17	13.12
			14	9.36	11.72
			40	10.43	10.43
	Without Action Tag	All Samples	5	9.61	11.21
			14	13.38	13.34
			40	11.64	11.57
Only PPC BB Coordinates	with Action Tag	Only PPC BB Coordinates	5	5.87	8.03
			14	5.04	7.30
			40	4.76	4.88
	Without Action Tag	Only PPC BB Coordinates	5	5.75	7.14
			14	5.47	8.44
			40	5.86	8.26
All BB Coordinates	With Action Tag	All BB Coordinates	5	6.22	6.89
			14	5.57	8.71
			40	4.10	6.07
	Without Action Tag	All BB Coordinates	5	6.20	6.86
			14	5.36	6.32
			40	4.01	4.60

In Table 3.2, we present a comparison between our time to cross estimation models. We also present the pedestrian time to cross estimation models on different prior time steps.

According to RMSE, the smallest error is the best one. We achieved 9.36 RMSE tested on real values (grand truth) and 10.42 RMNS tested on detected values. This

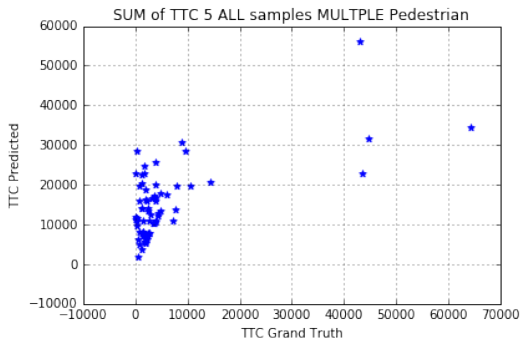
method was learned only on PPC samples with action tags and tested on all samples using 14 and respectively 40 frames as previous time step.

We obtained 4.76 RMSE tested on real values and 4.88 on detected values using only the PPC samples with action tags with 40 and respectively 5 frames as previous time steps.

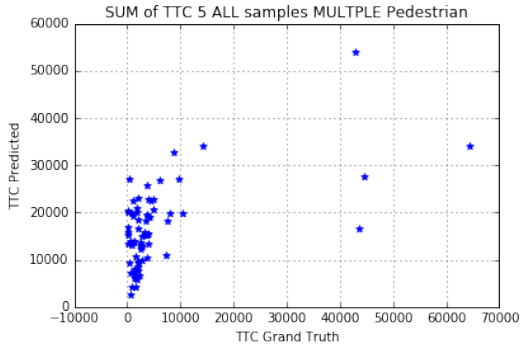
For the method learned on all BB coordinates samples we achieved the best performance with 40 frames at a time steps using only BB coordinates for both tested methods, (4.01 RMSE) real values and (4.60 RMSE) detected data methods.

We observed the best for those methods were obtained with different time steps. We think this difference comes from the various lengths of the pedestrian sequences and the complexity of data. However, the estimation of time to cross using all samples is more challenging for LTMS since it has to take into account even the pedestrian who are not preparing to cross the street, or whose intention is ambiguous.

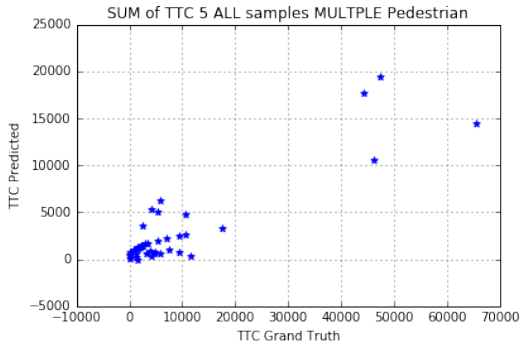
In Figures 3.13, 3.14 and 3.15 we plot the ground truth TTC versus predicted TTC on different time steps and for different approaches. We can observe that the estimation of TTC is globally satisfying. Indeed, the shape of the plot spread shows a roughly linear correlation between the real and the estimated values of TTC. We also plot the mean of RMNS values (see Figures 3.16, 3.17 and 3.18) for each video sequence on for different time steps and different approaches which confirms this observation. It confirms that the TCC values can be directly estimated in a regression method using a deep learning approach.



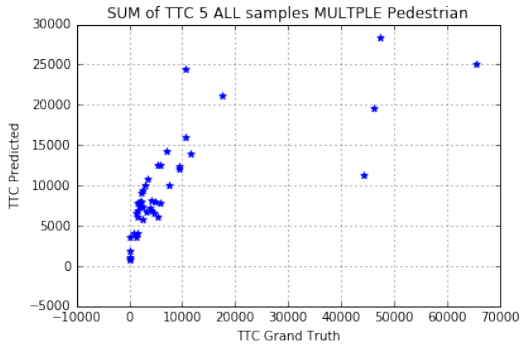
(a) TTC Predicted vs TTC Ground Truth real values on 5 time step, learned only on PPC samples with action tags and tested on the all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



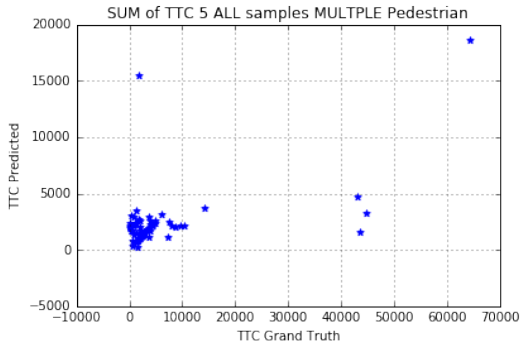
(b) TTC Predicted vs TTC Ground Truth detected values on 5 time step, learned only on PPC samples with action tags and tested on all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



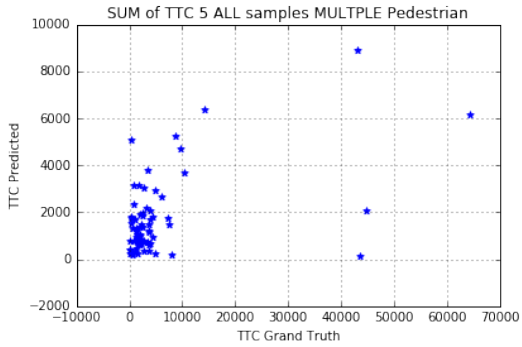
(c) TTC Predicted vs TTC Ground Truth real values on 5 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(d) TTC Predicted vs TTC Ground Truth detected values on 5 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

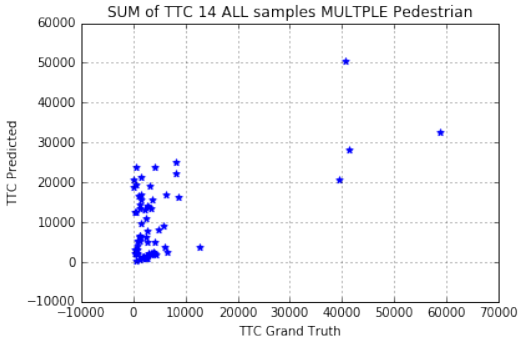


(e) TTC Predicted vs TTC Ground Truth real values on 5 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

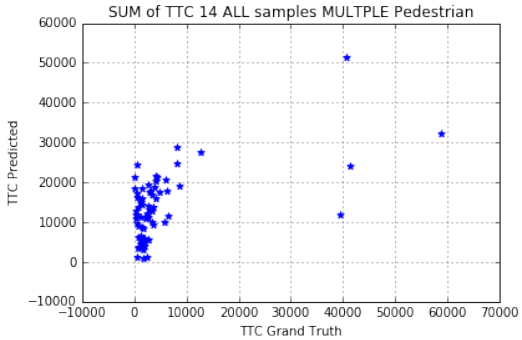


(f) TTC Predicted vs TTC Ground Truth detected values on 5 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

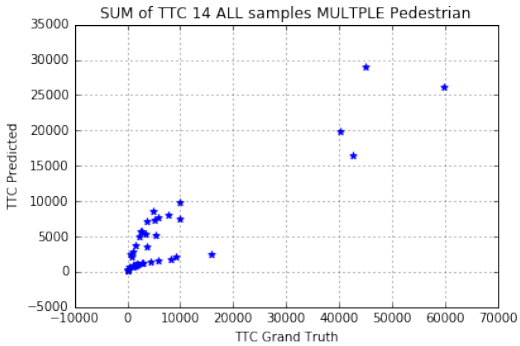
Figure 3.13 – Performance of the time to cross methods using 5 time step.



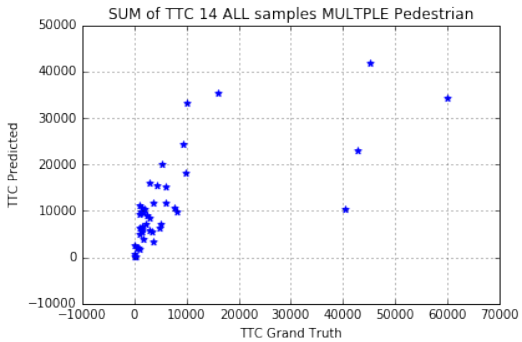
(a) TTC Predicted vs TTC Ground Truth real values on 14 time step, learned only on PPC samples with action tags and tested on the all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



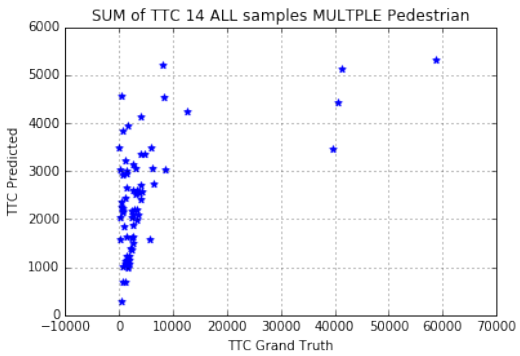
(b) TTC Predicted vs TTC Ground Truth detected values on 14 time step, learned only on PPC samples with action tags and tested on all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



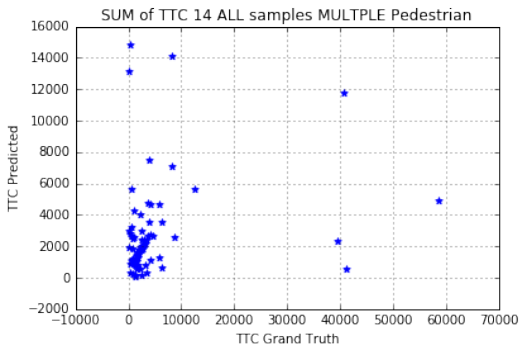
(c) TTC Predicted vs TTC Ground Truth real values on 14 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(d) TTC Predicted vs TTC Ground Truth detected values on 14 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

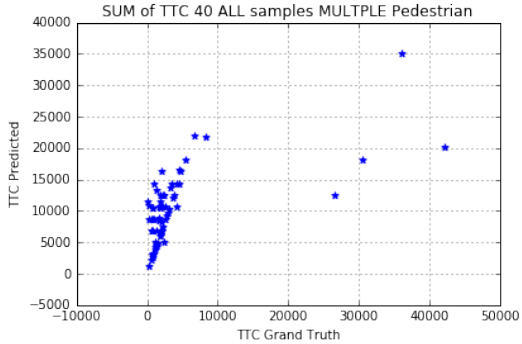


(e) TTC Predicted vs TTC Ground Truth real values on 14 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

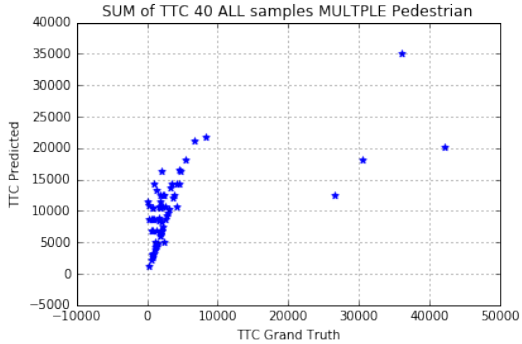


(f) TTC Predicted vs TTC Ground Truth detected values on 14 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

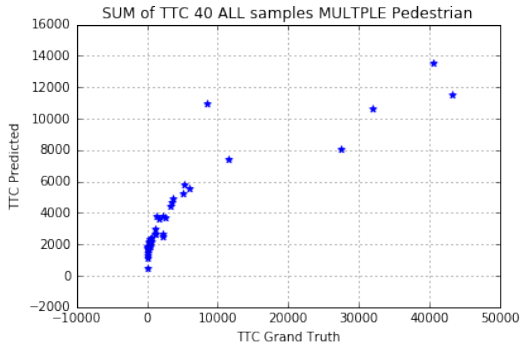
Figure 3.14 – Performance of the time to cross methods using 14 time step.



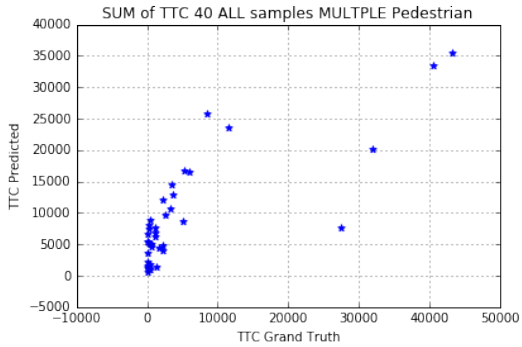
(a) TTC Predicted vs TTC Ground Truth real values on 40 time step, learned only on PPC samples with action tags and tested on the all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



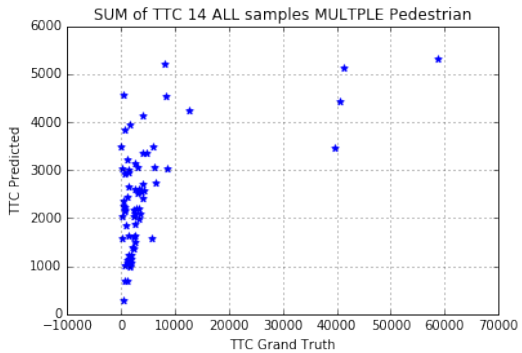
(b) TTC Predicted vs TTC Ground Truth detected values on 40 time step, learned only on PPC samples with action tags and tested on all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



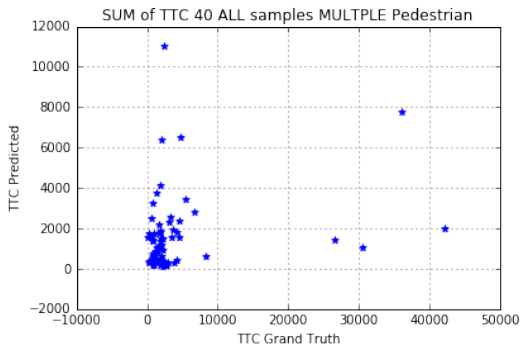
(c) TTC Predicted vs TTC Ground Truth real values on 40 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(d) TTC Predicted vs TTC Ground Truth detected values on 40 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

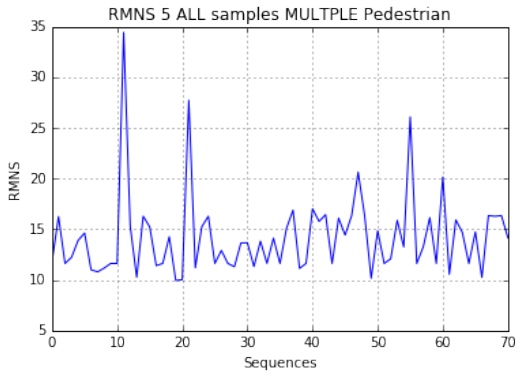


(e) TTC Predicted vs TTC Ground Truth real values on 40 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

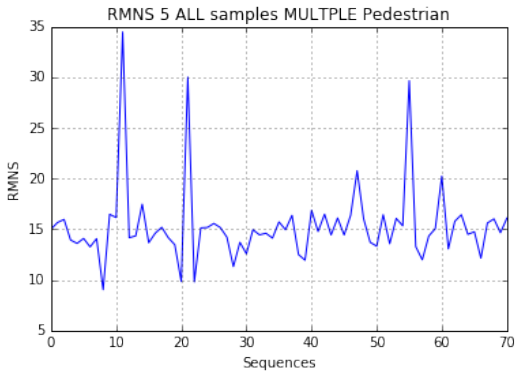


(f) TTC Predicted vs TTC Ground Truth detected values on 40 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

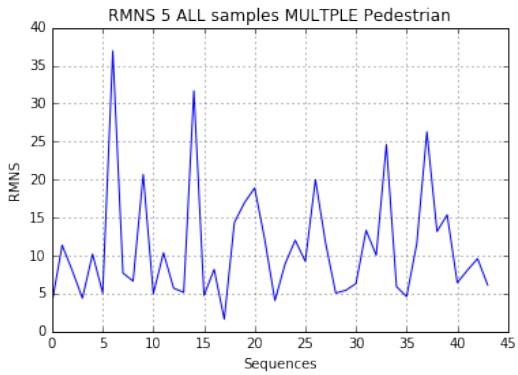
Figure 3.15 – Performance of the time to cross methods using 40 time step.



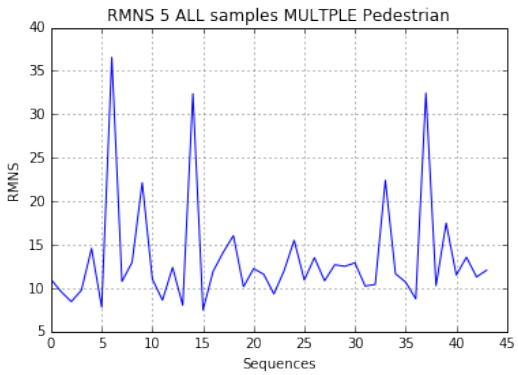
(a) RMNS on real values at 5 time step, learned only on PPC samples with action tags and tested on the all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



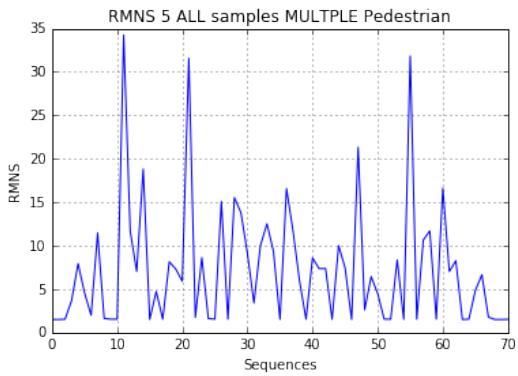
(b) RMNS on detected values at 5 time step, learned only on PPC samples with action tags and tested on all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



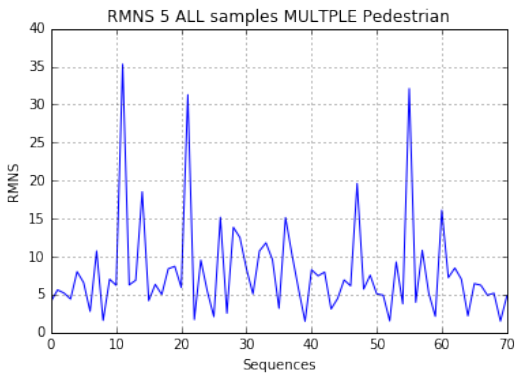
(c) RMNS on real values at 5 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(d) RMNS on detected values at 5 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

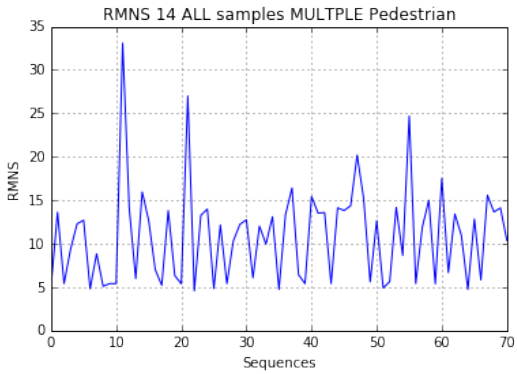


(e) RMNS on real values at 5 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

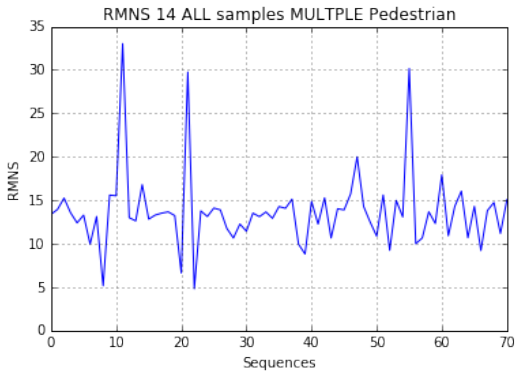


(f) RMNS on values at 5 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

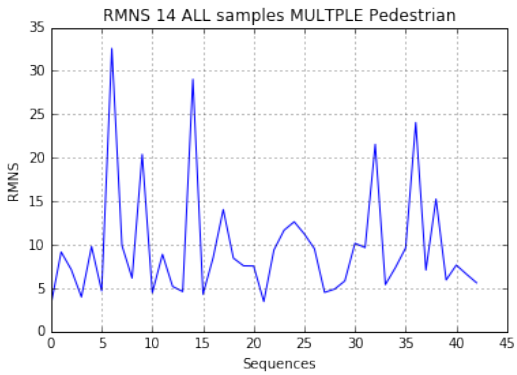
Figure 3.16 – RMNS performance of the time to cross methods using 5 time step.



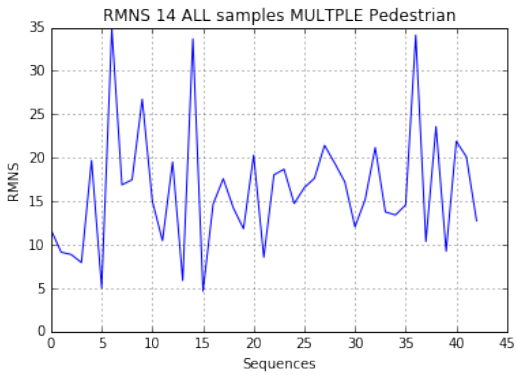
(a) RMNS on real values at 14 time step, learned only on PPC samples with action tags and tested on the all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



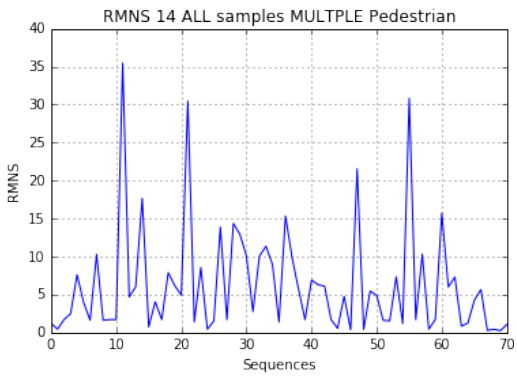
(b) RMNS on detected values at 14 time step, learned only on PPC samples with action tags and tested on all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



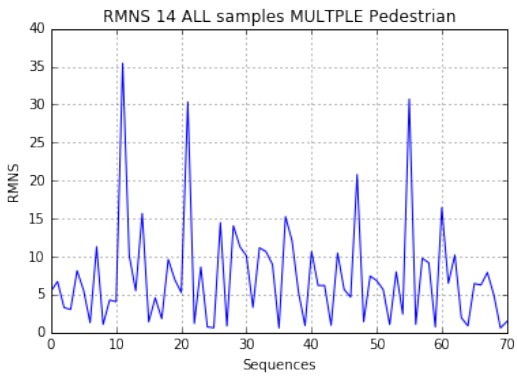
(c) RMNS on real values at 14 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(d) RMNS on detected values at 14 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

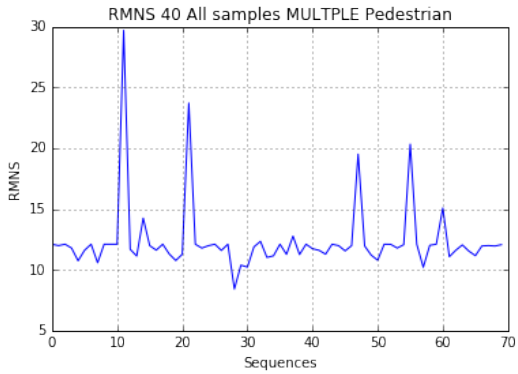


(e) RMNS on real values at 14 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

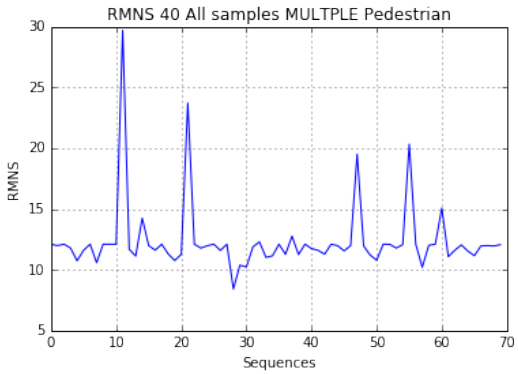


(f) RMNS on values at 14 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

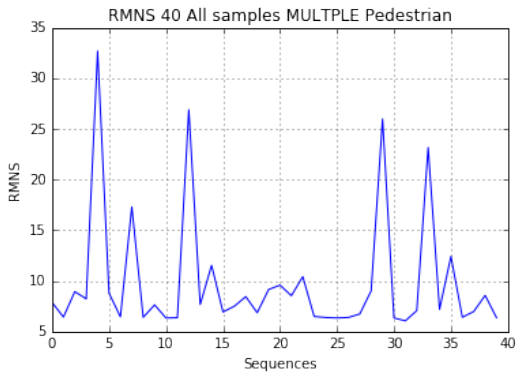
Figure 3.17 – RMNS performance of the time to cross methods using 14 time step.



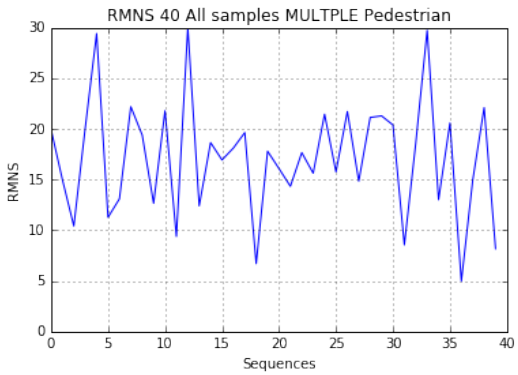
(a) RMNS on real values at 40 time step, learned only on PPC samples with action tags and tested on the all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



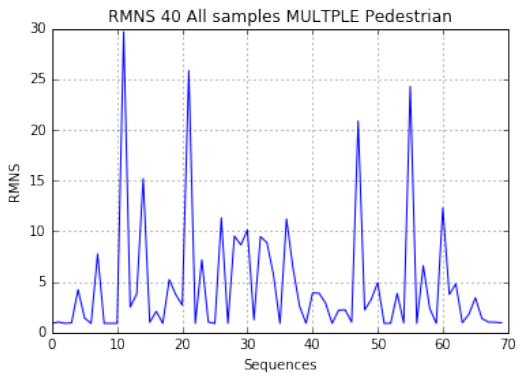
(b) RMNS on detected values at 40 time step, learned only on PPC samples with action tags and tested on all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



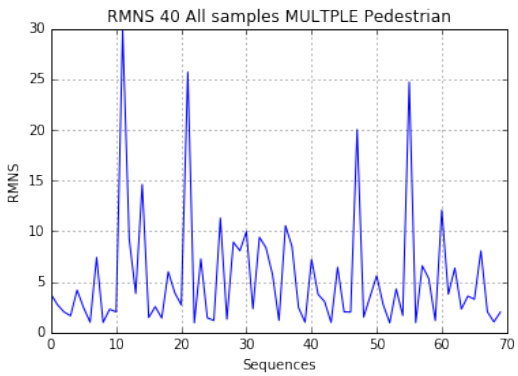
(c) RMNS on real values at 40 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(d) RMNS on detected values at 40 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(e) RMNS on real values at 40 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(f) RMNS on values at 40 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

Figure 3.18 – RMNS performance of the time to cross methods using 40 time step.

3.6 Conclusion

In this Chapter, we evaluated the estimation time to cross for pedestrians with deep learning approaches using the JAAD dataset.

We studied different pedestrian actions to find out if a pedestrian is crossing the street and based on this information, we estimate the time to cross for pedestrian. We split the pedestrian Joint Attention for Autonomous Driving (JAAD) dataset in into four classes: pedestrian is preparing to cross the street (PCC), the pedestrian is crossing the street (PC), pedestrian is about to cross the street (PAC), and pedestrian intention is ambiguous (PA).

We evaluated the pedestrian detection approach, where all samples are tagged as pedestrian and not pedestrian and a pedestrian detection approach using multiple tags. The first method achieved better performance since it has only to distinguish the pedestrians from other road users in contrast to the second one which has to recognize even the pedestrian actions. The second detection approach returned a weaker performance than the classical one.

The estimation of time to cross was learned using only PPC samples and all samples. Since our global method is created in two stages, the first one could be applied whenever the pedestrian detector returns the PPC event in contrast with the second one, which could be used without any restriction. The first one returns a better performance, but we consider the second one the more promising because it is more realistic, so we will continue to analyze it in our future our and also create an end-to-end detector-estimation time to cross approach.

Chapter 4

Conclusion

In this thesis, we have focused on developing a multi-task pedestrian protection system (PPS) which is an essential function of Advanced Driver Assistance systems (ADAS) because it reduces traffic accidents by assisting the driver and even stopping the vehicle to prevent imminent accidents. Our PPS system includes not only pedestrian classification, detection and tracking, but also pedestrian action-unit classification and prediction and, finally, pedestrian risk estimation (time to cross). This particular issue was solved by using original cross-modality deep learning approaches.

In Chapter 1, we introduced different learning methods based on Cross-Modality deep learning of Convolutional Neural Networks (CNNs) to solve the pedestrian classification issue:

- a Particular Cross-Modality learning method, where a CNN is trained and validated on the same image modality, but tested on a different one;
- a Separate Cross-Modality learning method which uses a different image modality for training than for validation;
- a Correlated Cross-Modality learning method where a unique CNN is trained and validated with Intensity, Depth and Optical Flow images for each frame;
- an Incremental Cross-Modality learning where a CNN is learnt with the first images modality frames, then a second CNN, initialized by transfer learning on the first CNN, is learnt on the second image modality frames, and finally a third CNN initialized on the second CNN, is learnt on the last image modality frames.
- an improvement of incremental cross-modality learning thanks a new CNN architecture that we proposed together with K-fold Cross-Validation of both the learning rate and epoch numbers.

We examine all these methods with the classical learning one where each CNN is trained and evaluated on the same image modality.

The experiments showed that the incremental cross-modality deep learning of CNNs achieves the best performances (distinguishing pedestrians and non pedestrians). It improves the classification performances not only for each modality classifier but also for the multi-modality late-fusion scheme. We analyzed the incremental cross modality deep learning even in the second part of our research.

In Chapter 2, we addressed severs problems:

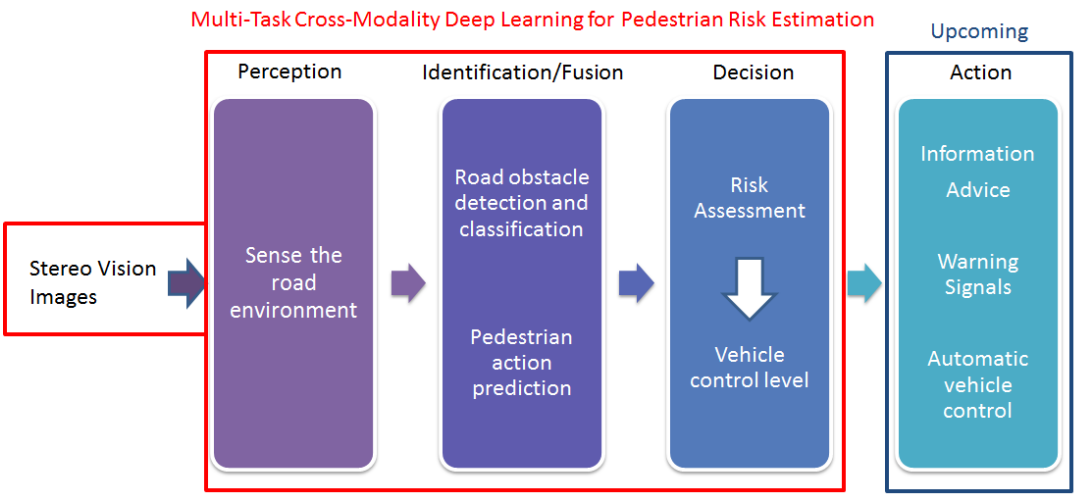


Figure 4.1 – The main architecture of our upcoming system.

- We applied the incremental cross-modality deep learning method on the detection method;
- We found out whether a pedestrian is crossing, and whether the pedestrian's action does not present a critical situation, where we have defined four main pedestrian actions:
  1. the pedestrian is preparing to cross the street;
  2. the pedestrian is crossing the street;
  3. the pedestrian is about to cross the street;
  4. the pedestrian's intention is ambiguous;
- We introduced a unified pedestrian detection component based on incremental cross-modality deep learning, which also recognizes different pedestrian actions.

The incremental cross-modality deep learning method outperformed the classical detection approach on all modalities, but its performance is statistically significant only for the RGB image modality. We noticed that the performance of the incremental cross-modality deep learning detector is directly proportional to the achievements of each detection of pedestrian actions. We validated the Incremental Cross-Modality learning method not only for pedestrian classification, but also for pedestrian unit action recognition and pedestrian detection.

We extended the pedestrian detection component using incremental cross modality deep learning by taking into account the temporal context in order to predict the next pedestrian action. We analyzed this issue in the third part of our research without using the incremental cross-modality deep learning.

In Chapter 3, we merged the pedestrian detection component with the pedestrian action prediction and estimation of time to cross.

We developed a prediction of pedestrian action using an estimation of time to cross for a single and multiple pedestrians using a Long Short-Term Memory (LSTM)

We used a Long Short-Term Memory (LSTM) [HS97] to estimate the pedestrian intention action using the previous 5, 14, and respectively 40 frames as time steps.

We showed that integrating multiple pedestrian tags for the detection part and merging with LSTM, can achieve a significant performance.

Since in our thesis we have managed to develop only the first three-component (Perception, Identification/Fusion, and Decision) from our main project, for last component (Actions), we will implement it in the Inria's vehicle in our future work (see Figure 4.1). More then that, for continuing the work, we are planning to create an end-to-end incremental cross-modality deep leaning detector-estimation time to cross approach, which will be able to do all the functionalities in one step (detection, action recognition, action prediction, estimation of time to cross). In addition, we intend to apply the incremental cross-modality deep leaning model for the classification and detection of other road objects (traffic signs and traffic lights) as well as road users (vehicles, cyclists).

Appendix A

Annexes

A.1 RTMAPS Architecture

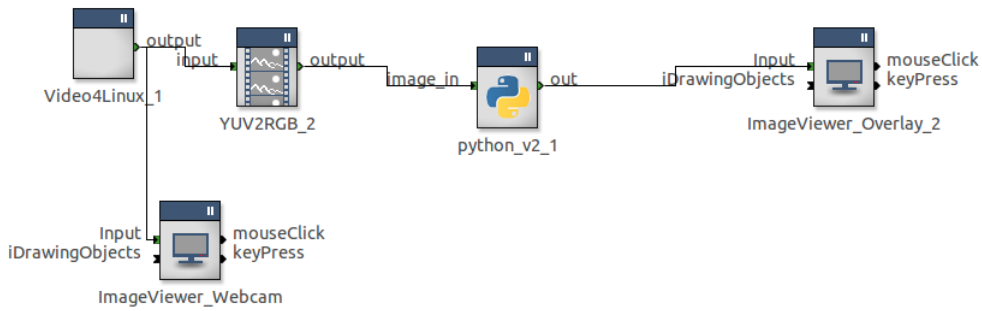


Figure A.1 – The RTMAPS Detection Architecture for RGB

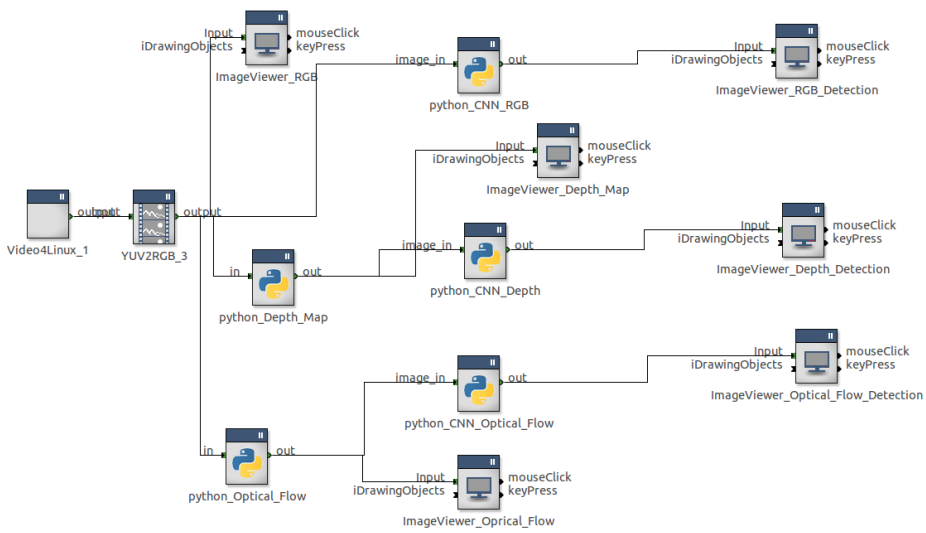


Figure A.2 – The RTMAPS Detection Architecture for multiple image modality

Bibliography

[AGR+16] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, June 2016. 78

[AH16] Ali Ismail Awad and Mahmoud Hassaballah. *Image Feature Detectors and Descriptors: Foundations and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2016. 16

[ATI19] F. Ahmed, B. A. Topu, and S. M. M. Islam. Hog and gabor filter based pedestrian detection using convolutional neural networks. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–6, Feb 2019. 15

[AWLJ16] Abrar H. Abdulnabi, Gang Wang, Jiwen Lu, and Kui Jia. Multi-task CNN model for attribute prediction. *CoRR*, abs/1601.00400, 2016. 49

[BDX16] R. Bunel, F. Davoine, and Philippe Xu. Detection of pedestrians at far distance. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2326–2331, May 2016. 15, 48

[BFF+19] S. Baabou, A. B. Fradj, M. A. Farah, A. G. Abubakr, F. Bremond, and A. Kachouri. A comparative study and state-of-the-art evaluation for pedestrian detection. In *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pages 485–490, March 2019. 48

[BJNL13] Sheryl Brahnam, Lakhmi C. Jain, Loris Nanni, and Alessandra Lumini. *Local Binary Patterns: New Variants and Applications*. Springer Publishing Company, Incorporated, 2013. 46

[BKFG19] M. Braun, S. Krebs, F. Flohr, and D. Gavrila. Eurocity persons: A novel benchmark for person detection in traffic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 48, 54, 78

[BMM11] L. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *2011 International Conference on Computer Vision*, pages 1543–1550, Nov 2011. 49

[BOHS14] Rodrigo Benenson, Mohamed Omran, Jan Hendrik Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned? *CoRR*, abs/1411.4304, 2014. 14, 46, 78

[Bot12] Léon Bottou. Stochastic gradient descent tricks. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade: Second Edition*, pages 421–436, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 20, 25, 27

[BT10] Alexander Borichev and Yuri Tomilov. Optimal polynomial decay of functions and operator semigroups. *Mathematische Annalen*, 347(2):455–478, Jun 2010. 20, 27

[CJG+15] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032, Dec 2015. 17

[CS14] W. Choi and S. Savarese. Understanding collective activitiesof people from videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1242–1257, June 2014. 75

[CvMG+14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. 79

[DH92] J. P. DeCruyenaere and H. M. Hafez. A comparison between kalman filters and recurrent neural networks. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 4, pages 247–251 vol.4, June 1992. 82

[DHS10] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Technical Report UCB/EECS-2010-24, EECS Department, University of California, Berkeley, Mar 2010. 20

[DLHS16] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. *CoRR*, abs/1605.06409, 2016. 46, 47, 85

[DT05a] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005. 46, 48, 54

[DT05b] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society. 14

[DTPB09] Piotr Dollar, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. In *Proc. BMVC*, pages 91.1–91.11, 2009. doi:10.5244/C.23.91. 14, 46

[DWSP09] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009. 6, 18, 27, 48, 54

[DWSP12a] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):743–761, April 2012. 14

[DWSP12b] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):743–761, April 2012. 14

[EESG10] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 990–997, June 2010. 6, 12, 15, 17, 19, 20, 22, 26, 29, 33, 35, 36, 37, 38, 43, 75

[EG09] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, Dec 2009. 54

[EG11] M. Enzweiler and D. M. Gavrila. A multilevel mixture-of-experts framework for pedestrian classification. *IEEE Transactions on Image Processing*, 20(10):2967–2979, Oct 2011. 15, 19, 22, 26, 34, 36, 37, 75

[ER16] M. Errami and M. Rziza. Improving pedestrian detection using support vector regression. In *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pages 156–160, March 2016. 14

[ESWG16] M. Eisenbach, D. Seichter, T. Wengefeld, and H. M. Gross. Cooperative multi-scale convolutional neural networks for person detection. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 267–276, July 2016. 15, 48

[FGMR10a] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010. 14

[FGMR10b] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010. 18, 46

[FL18] Z. Fang and A. M. López. Is the pedestrian going to cross? answering by 2d pose estimation. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1271–1276, June 2018. 78, 81, 82

[FYY+15] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, and H. Murase. Pedestrian detection based on deep convolutional neural network with ensemble inference network. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 223–228, June 2015. 15

[GAG+17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 79

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010. 20

[GDDM13] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013. 46

[GGM14] Georgia Gkioxari, Ross B. Girshick, and Jitendra Malik. Actions and attributes from wholes and parts. *CoRR*, abs/1412.2604, 2014. 49

[GHW18] L. Gong, W. Hong, and J. Wang. Pedestrian detection algorithm based on integral channel features. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 941–946, June 2018. 14

[Gir15] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society. 43, 46, 47, 48

[GLSU13] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 48, 54

[HJ15a] J. Hariyono and K. Jo. Detection of pedestrian crossing road. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4585–4588, Sep. 2015. 50

[HJ15b] J. Hariyono and K. Jo. Pedestrian action recognition using motion type classification. In *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*, pages 129–132, June 2015. 78

[HK16] J. Hariyono and Kang-Hyun Jo. Centroid based pose ratio for pedestrian action recognition. In *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, pages 895–900, June 2016. 50

[HOBS15] Jan Hosang, Mohamed Omran, Rodrigo Benenson, and Bernt Schiele. Taking a deeper look at pedestrians. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 15

[HPK+15] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 54

[HRS+16a] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016. 45, 51

[HRS+16b] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016. 47

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. 8, 77, 79, 104

[HTDD18] M. Hoy, Z. Tu, K. Dang, and J. Dauwels. Learning to predict pedestrian intention via variational tracking networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3132–3137, Nov 2018. 78, 81, 83

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 43, 47, 51

[IC14] Ozan İrsoy and Claire Cardie. Deep recursive neural networks for compositionality in language. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2096–2104, Cambridge, MA, USA, 2014. MIT Press. 79

[JSD+14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 27

[JWZ13] J. Joo, S. Wang, and S. Zhu. Human attribute recognition by rich appearance dictionary. In *2013 IEEE International Conference on Computer Vision*, pages 721–728, Dec 2013. 49

[JZ15] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, Denver, Colorado, May–June 2015. Association for Computational Linguistics. 79, 80

[KAHS16] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto. Intent-aware long-term prediction of pedestrian motion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2543–2549, May 2016. 75

[KF18] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *CoRR*, abs/1806.11230, 2018. 49

[KG14] C. G. Keller and D. M. Gavrila. Will the pedestrian cross? a study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):494–506, April 2014. 81

[KG16] C. Karaoguz and A. Gepperth. Incremental learning for bootstrapping object classifier models. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1242–1248, Nov 2016. 16

[KML+18] T. Kim, M. Motro, P. Lavieri, S. S. Oza, J. Ghosh, and C. Bhat. Pedestrian detection with simplified depth prediction. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2712–2717, Nov 2018. 48

[KRT16] Iuliia Kotseruba, Amir Rasouli, and John K. Tsotsos. Joint attention in autonomous driving (JAAD). *CoRR*, abs/1609.04741, 2016. viii, 7, 44, 45, 51, 54, 55, 56, 57, 76, 77, 81, 82, 86

[KSH12a] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 16, 18, 25, 27, 43, 46

[KSH12b] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 20

[LAE+15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. 43, 46, 47, 48, 85

[LBBH98] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. 15, 17, 18, 20, 25, 27, 31

[LBH15] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. 17

[LCH15] D. Li, X. Chen, and K. Huang. Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 111–115, Nov 2015. 49

[LCZH18] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. Pose guided deep model for pedestrian attribute recognition in surveillance scenarios. In *2018 IEEE International Conference on Multimedia and Expo, ICME 2018, San Diego, CA, USA, July 23-27, 2018*, pages 1–6, 2018. 49

[LDWW18] W. Lan, J. Dang, Y. Wang, and S. Wang. Pedestrian detection based on yolo network model. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1547–1551, Aug 2018. 48, 78

[LGG+17a] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. 45, 47, 51, 59, 61, 77, 82, 85, 88

[LGG+17b] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. 47

[LHLT16] Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. Human attribute recognition by deep hierarchical contexts. In *European Conference on Computer Vision*, 2016. 49

[LL16] S. Lin and C. Lee. Pedestrians and vehicles recognition based on image recognition and laser distance detection. In *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pages 1232–1237, Oct 2016. 78

[LLYS18] Pengze Liu, Xihui Liu, Junjie Yan, and Jing Shao. Localization guided learning for pedestrian attribute recognition. *CoRR*, abs/1808.09102, 2018. 49

[LYT16] Yichen Wei Shuang Liang Luwei Yang, Ligeng Zhu and Ping Tan. Attribute recognition from adaptive parts. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 81.1–81.11. BMVA Press, September 2016. 49

[LZT+17] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 49

[MU08] R. Matei and P. Ungureanu. A class of gaussian-shaped cnn filter banks. In *2008 11th International Workshop on Cellular Neural Networks and Their Applications*, pages 135–139, July 2008. 20

[NGB17] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs non-handcrafted features for computer vision classification. *Pattern Recognition*, 71, 06 2017. 16

[OW12] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3258–3265, June 2012. 15, 19, 36, 37

[OZL+17] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, and X. Wang. Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017. 15, 48

[PESv09] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, Sep. 2009. 75

[PGD+17] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 53

[PP17] S. S. Patil and P. Palanisamy. Pedestrian classification in partial occlusion. In *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, pages 1–6, March 2017. 14

[PRNB17] Dănuţ Ovidiu Pop, Alexandrina Rogozan, Fawzi Nashashibi, and Abdelaziz Bensrhair. Incremental cross-modality deep learning for pedestrian recognition. *In 28th IEEE Intelligent Vehicles Symposium (IV)*, pages 523–528, June 2017. 78

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 27, 91

[QPLS14] R. Quintero, I. Parra, D. F. Llorca, and M. A. Sotelo. Pedestrian path prediction based on body language and action classification. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 679–684, Oct 2014. 78

[RDGF15] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. 43, 46, 47, 85

[RHGS15a] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 43, 46, 47, 51, 85

[RHGS15b] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. 48

[RK15a] E. Rehder and H. Kloeden. Goal-directed pedestrian prediction. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 139–147, Dec 2015. 78, 83

[RK15b] Eike Rehder and Horst Kloeden. Goal-directed pedestrian prediction. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015. 75

[RKT17a] Amir Rasouli, Iuliia Kotseruba, and John K. Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 51, 54, 65, 81

[RKT17b] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–213, 2017. 51, 55, 86

[RKT18] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Understanding pedestrian behavior in complex traffic scenes. *IEEE Transactions on Intelligent Vehicles*, 3(1):61–70, March 2018. 51, 55, 86

[RR19] N. K. Ragesh and R. Rajesh. Pedestrian detection in automotive safety: Understanding state-of-the-art. *IEEE Access*, 7:47864–47890, 2019. 48

[RRL+18] D. Ridel, E. Rehder, M. Lauer, C. Stiller, and D. Wolf. A literature review on the prediction of pedestrian behavior in urban scenarios. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3105–3112, Nov 2018. 78, 81, 83

[RSZS16] R. Rauf, A. R. Shahid, S. Ziauddin, and A. A. Safi. Pedestrian detection using hog, luv and optical flow as features with adaboost as classifier. In *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–4, Dec 2016. 14

[RWLS18] E. Rehder, F. Wirth, M. Lauer, and C. Stiller. Pedestrian prediction by planning using deep neural networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–5, May 2018. 78, 81, 83

[Sch15] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. Published online 2014; based on TR arXiv:1404.7828 [cs.NE]. 17

[SCK16] J. Schlosser, C. K. Chow, and Z. Kira. Fusing lidar and images for pedestrian detection using convolutional neural networks. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2198–2205, May 2016. 78

[SG13a] Nicolas Schneider and Dariu M. Gavrila. Pedestrian path prediction with recursive bayesian filters: A comparative study. In Joachim Weickert, Matthias Hein, and Bernt Schiele, editors, *Pattern Recognition*, pages 174–183, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 50, 86

[SG13b] Nicolas Schneider and Dariu M. Gavrila. Pedestrian path prediction with recursive bayesian filters: A comparative study. In Joachim Weickert, Matthias Hein, and Bernt Schiele, editors, *Pattern Recognition*, pages 174–183, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 78, 81, 83

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014. 20

[SKA+17] Muhammad Sharif, Muhammad Attique Khan, Tallha Akram, Muhammad Younus Javed, Tanzila Saba, and Amjad Rehman. A framework of human detection and action recognition based on uniform segmentation and combination of euclidean distance and joint entropy-based

features selection. *EURASIP Journal on Image and Video Processing*, 2017(1):89, Dec 2017. 50

[SKHD09] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. In *2009 IEEE 12th International Conference on Computer Vision*, pages 24–31, Sept 2009. 14

[SLJ+14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 15, 25, 43

[SSL15] Patrick Sudowe, Hannah Spitzer, and Bastian Leibe. Person attribute recognition with a jointly-trained holistic cnn model. In *ICCV Workshops*, pages 329–337. IEEE Computer Society, 2015. 49

[Sun13] Peng Sun. Exponential decay of expansive constants. *Science China Mathematics*, 56(10):2063–2067, Oct 2013. 20, 27

[SXK18] Nikolaos Sarafianos, Xiang Xu, and Ioannis A. Kakadiaris. Deep imbalanced attribute classification using visual attention aggregation. *CoRR*, abs/1807.03903, 2018. 49

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 15, 18, 25, 43

[TG12] Tieleman T. and Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. In *COURSERA: Neural Networks for Machine Learning*, 2012. 20, 27

[VGVZ09a] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009. 14

[VGVZ09b] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009. 46

[VLM+14] David Vazquez, Antonio M. Lopez, Javier Marin, Daniel Ponsa, and David Geronimo. Virtual and real world adaptation for pedestrian detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 36(4):797–809, 2014. 16

[WBSS04] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. 27, 29

[WFHB16] Jörg Wagner, Volker Fischer, Michael Herman, and Sven Behnke. Multispectral pedestrian detection using deep fusion convolutional neural networks. In *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 509–514, April 2016. 16

[WXY16] L. Wang, L. Xu, and M. Yang. Pedestrian detection in crowded scenes via scale and occlusion analysis. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1210–1214, Sep. 2016. 15

[WZY+19] Xiao Wang, Shaofei Zheng, Rui Yang, Bin Luo, and Jin Tang. Pedestrian attribute recognition: A survey. *CoRR*, abs/1901.07474, 2019. 49

[XWK+15] Xiaogang, Pengxu Wei, Wei Ke, Qixiang Ye, and Jianbin Jiao. Pedestrian detection with deep convolutional neural network. In C.V. Jawahar and Shiguang Shan, editors, *Computer Vision - ACCV 2014 Workshops: Singapore, Singapore, November 1-2, 2014, Revised Selected Papers, Part I*, pages 354–365, Cham, 2015. Springer International Publishing. 15, 48

[YML17] B. Yu, Y. Ma, and J. Li. Fast pedestrian detection with multi-scale classifiers. In *2017 International Conference on Computing Intelligence and Information System (CIIS)*, pages 225–230, April 2017. 15

[ZBO+16] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hendrik Hosang, and Bernt Schiele. How far are we from solving pedestrian detection? *CoRR*, abs/1602.01237, 2016. 46, 78

[ZLLH16] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 443–457, Cham, 2016. Springer International Publishing. 48, 61

[ZPR+13] Ning Zhang, Manohar Paluri, Marc'Aurelio Ranzato, Trevor Darrell, and Lubomir D. Bourdev. PANDA: pose aligned networks for deep attribute modeling. *CoRR*, abs/1311.5591, 2013. 49