Pedestrian movement direction recognition using convolutional neural networks

Alex Dominguez-Sanchez, Miguel Cazorla, *Senior Member, IEEE*, Sergio Orts-Escolano

***Abstract*—Pedestrian movement direction recognition is an important factor in autonomous driver assistance and security surveillance systems. Pedestrians are the most crucial and fragile moving objects in streets, roads and events where thousands of people may gather on a regular basis. People flow analysis on zebra crossings and in shopping centres or events such as demonstrations are a key element to improve safety and to enable autonomous cars to drive in real life environments. This paper focuses on deep learning techniques such as Convolutional Neural Networks (CNN) to achieve a reliable detection of pedestrians moving in a particular direction. We propose a CNN-based technique that leverages current pedestrian detection techniques (HOG-linSVM) to generate a sum of subtracted frames (flow estimation around the detected pedestrian), which are used as an input for the proposed modified versions of various state-of-the-art CNN networks such as AlexNet, GoogleNet and ResNet. Moreover, we have also created a new dataset for this purpose, and analysed the importance of training in a known dataset for the neural networks to achieve reliable results.**

***Index Terms*—pedestrian detection, advance driver assistance system, convolutional neural networks, pedestrian intention recognition**

I. Introduction

TRAFFIC control, risk detection and Autonomous Driver Assistance Systems (ADAS) are key elements for the development of future intelligent transportation systems. Furthermore, dynamic pedestrian movement in traffic environments makes it necessary to develop people flow analysis and movement intention recognition systems. In recent years, Convolutional Neural Networks (CNN) and other deep learning techniques have demonstrated impressive performance in many computer vision problems and therefore we believe they could be the perfect approach for the aforementioned problems. Moreover, computer vision and machine learning techniques have been transformed due to the rapid evolution and remarkable performance of Graphics Processing Units (GPUs), which has enabled the development of deep learning-based systems. In this work, our objective is the detection and recognition of pedestrian intention on streets, zebra crossings or road junctions, so as to be able to alert drivers or monitoring systems about possible risk situations.

A. Areas of interests

What is needed to classify the images from a video to be able to distinguish if the pedestrian is moving towards the left or the right? Image classification and, consequently, Convolutional Neural Networks (CNN) in computer vision, have become a popular subject of research. In this work, we will focus on CNNs as a tool for our recognition problem together with Histograms of Oriented Gradients (HOG) and other pixel-based techniques for dataset creation.

B. Related works

Until 2012, most recognition, segmentation and classification image problems were approached by extracting hand-designed features and applying specific algorithms for those particular features. For example, if a number plate on a car needed to be detected, we segmented the image by looking for straight lines, then corners and finally the image was reduced until we had an area similar to the geometry of a number plate. In essence, we looked for the particular features that could solve a specific problem.

A common hand-crafted feature used for pedestrian detection is the Histogram of Oriented Gradients (HOG) [1]. The main idea behind this descriptor is that local object appearance and shape within an image can be described by the intensity distribution of gradients or edge directions. The image is divided into small connected areas, and for the pixels within each area, a histogram of gradient directions is generated. Finally, the descriptor is the concatenation of these histograms.

Recent work in this area has added a local sub-descriptor called Colour Self Similarity (CSS) [2] where colour histograms are compared within a HOG detected window, and for example, colour histograms from the two arms have a high similarity.

In addition, extensive research has been done on pedestrian detection [3], [4], where more than sixteen different detectors were benchmarked [5] against several public datasets. Most of these hand-designed features were studied in [5]. The features were mainly based on window-sliding techniques and detection was performed using support vector machines (SVM) for classification. Moreover, other approaches based on the Adaboost work of Viola and Jones [6], and many others based on HOG and variants of the same method [7], were extensively evaluated in [5].

Since 2012, new approaches for pedestrian detection and related problems have emerged with the advent of deep learning techniques. Deep learning is a new way of applying machine learning algorithms, where neural networks are being made deeper and deeper by the addition of tens, or even hundreds, of layers. Specifically, in computer vision, much work was done in this regard before 2012, using multi-layer neural networks but obtaining poor results. Recognition of characters was conducted using a CNN [8] with a deeper layer structure. However, it was after 2012, with the proposal of Alex Krizhevsky CNN, AlexNet, when the real capabilities

of CNNs became clear. These methodologies were first used at the Imagenet Competition [9] where the novel techniques, of deep multi-layer neural networks, were accelerated using GPUs. Since then, new and better hardware has appeared. This increases the possibility of bigger and deeper CNNs, providing better classification accuracy and making the training of existing deep networks an affordable scientific tool in terms of training time.

Computer vision research groups focused on pedestrian detection have also benefited from the rise of CNNs, and recent analyses have proved that better and more reliable results can be achieved [10]. However, our work focuses, not just on pedestrian detection, but also on pedestrian movement direction recognition, analyzing, for example, whether the pedestrian moved to the left, right or to the front of the scene. There are few studies in this area. Enzweiler and Gavrila [11] and Gandhi et al. [12] focused on that aspect using the HOG descriptor and SVM as a classifier while Mogelmose et al. [13] used pedestrian tracking techniques and trajectory analysis for estimating pedestrian direction.

In general, the estimation of pedestrians' trajectories have traditionally been addressed using naive movement models based on human gait estimation and analysis of simple heuristics based on that information [14], [15]. Other traditional approaches have focused on the use of Kalman Filters (KF) to estimate pedestrian trajectories. Most of these existing techniques produced poor results due to the impossibility of properly handling and adapting to changes in pedestrians' movements [16]. More recently, a more complex method based on Artificial Neural Networks (ANN) has been proposed for pedestrian trajectory estimation and intention recognition [15]. This work is able to estimate pedestrian trajectory based on pedestrian head detection and the use of its position for tracking along the sequence. Other existing works in the literature make use of features extracted from a dense optical compensated with ego-motion techniques (car movement) [17]. Using this approach, they are not only able to estimate a pedestrian's path but also to roughly estimate pedestrian intentions towards specific situations such as crossing at intersections [18].

Finally, it is worth mentioning the existence of related works addressing this problem from a different perspective. Most of these works are based on the information gathered by inertial measurement units (IMUs) and similar technologies (accelerometers, gyroscopes, etcetera). These types of approaches are very intrusive from the pedestrian viewpoint and do not provide enough information to distinguish between different pedestrians' actions.

After reviewing state-of-the-art techniques we can conclude that even though in recent years great progress has been made in pedestrian recognition systems, more research is still required on systems and new techniques that can provide better classification accuracy, improved performance and ease of integration in current ADAS and security surveillance systems.

In this work, we contribute to the literature on pedestrian walking direction recognition with the proposal of a modified CNN-based system, which has been tested on different CNNs architectures such as *AlexNet* [9], *GoogleNet* [19] and *ResNet* [20]. These CNNs have been trained with a novel dataset that was recorded in different scenarios. Pedestrians were video recorded and the CNNs were fed with output images produced as a result of several image operations at pixel level from this input video. The main purpose of this additional image processing was to visually highlight image characteristics that may be relevant for pedestrian trajectory recognition.

Videos showing the processing pipeline and the proposed dataset are available on our project website[1].

To the best of our knowledge, no work has been done on the classification of pedestrians according to their motion direction using deep learning techniques. The key contributions of our work are as follows:

- We provide a new dataset containing ground truth information for pedestrian movement direction on urban roads.
- We propose a novel pipeline for pedestrian movement direction recognition, which provides high recognition rates in the proposed dataset.
- We have evaluated state-of-the-art Convolutional Neural Network models for the problem presented and carried out a performance evaluation providing quantitative metrics such as accuracy and execution time.
- We have developed an implementation of the proposed pipeline that runs at 18 Hz on commodity GPU hardware.

The rest of the paper is organized as follows: Section II, we briefly present convolutional neural networks, and describe various architectures that have been used in this work. Section III presents and justifies the need for a new dataset for pedestrian trajectory estimation. In Section IV, we describe the proposed CNN-based method for pedestrian movement direction estimation. Section V explains how we have trained and evaluated our CNN-based system in several experiments, including a hyper-parameter search to find optimum parameters for training the proposed network. It also presents the results obtained (accuracy and runtime) for the three evaluated CNN architectures: AlexNet, GoogLeNet and ResNet. Finally, Section VI draws conclusions and indicates lines of future research work.

II. Convolutional Neural Networks

A Convolutional Neural Network is very similar to a standard Neural Network. This kind of network consists of neurons with learnable weights. Each neuron receives inputs from the input data, and performs a dot product with some initial weights that will be modified during the training process (backpropagation). The network expresses a single differentiable score function, the gradient, from the raw image pixels, and outputs a classification class as a result. They also have a loss function that is used to minimize the score.

A CNN exploits image structural information and builds the neural network model in a more intelligent way than conventional neural networks. In particular, unlike a normal neural network, the layers of a CNN have neurons organized in three dimensions: x, y, z, instead of 2D vectors. In Figure 1, we see an example, where the input, the pink layer, has width x, height y and depth: RGB channels. The blue layers

[1] http://www.rovit.ua.es/dataset/pedirecog/

are the hidden ones, convoluted by a given filter (e.g., 3x3). There are also 3D or n-dimensional layers. The neurons of all layers are activated using the ReLU (Rectified Linear Unit) activation function, where given an input value z, the ReLU layer computes the output as z if $z > 0$, with no negative slope.

In particular, in this work we have used various modified versions of state-of-the-art CNN networks. The first CNN network we evaluated was AlexNet, which has five convolutional layers and three fully connected layers. The last layer only contains 3 neurons with a Softmax function, able to recognize the three classes of pedestrians trajectories. Moreover, we also trained two additional CNN architectures: GoogLeNet and ResNet.

GoogLeNet is a CNN architecture for image recognition tasks, providing the highest accuracy in several challenges. In this architecture, all convolutions, including those inside the inception modules, use ReLU activation. The size of the receptive field in this network is 224×224 taking RGB channels. GoogLeNet network is 22 layers deep when considering only layers with parameters (or 27 layers if we also consider pooling ones). The overall number of layers (independent building blocks) used for the construction of the network is about 100. The use of average pooling step prior to the classifier is based on [21], although this implementation differs in the use of an extra linear layer. This enables adapting and fine-tuning the network for other datasets.

The ResNet50 deep learning architecture is currently the state-of-the-art CNN for image recognition, achieving a top-1 error (22.85) on the ImageNet validation split [22]. This architecture introduces the "residual" term, which consists in the aggregation of the input image to the output image of a convolution block. As a result, the output of a convolution block can be seen as the input image where the features activated by the filters are highlighted. In contrast, the output of a convolution layer in a default Convolutional Neural Network is only the result of neuron activation. If a neuron is not triggered on a certain region of the input image, the activation values of the output remains low. When the network computes the weights update in the backpropagation stage, the values in non-activated regions lead to very low updates, eventually even causing no update at all, which causes the learning to get stuck (also known as the vanishing gradient problem). The inclusion of the "residual" term helps fight the vanishing gradient problem and allows the design of even deeper architectures. Currently, the best performer on several tasks of the ImageNet challenge is based on the "residual" approach introduced by ResNet.

III. Dataset

In this section we present our pedestrian movement direction recognition dataset. This dataset is used for the training and evaluation of the proposed system. To test the proposed CNN-based system we needed a specific dataset, designed to feed our network with images of pedestrians moving in different directions and in different scenarios, boardwalks, zebra crossings, sidewalks, etc. Video was recorded with a camera

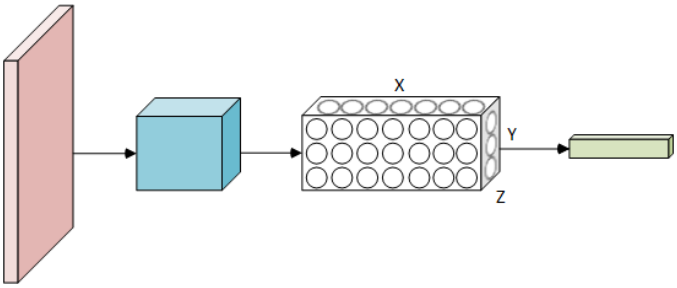

Fig. 1. Example of input image in a Convolutional Neural Network layer. 3D input layer, 3D hidden layers

Fig. 2. Dataset images: Left, right and front samples

capturing at 30fps, 640x480 resolution. Videos were recorded in a static and dynamic manner, using a regular and a hand-held tripod. To the best of our knowledge, there are no other datasets that provide information about pedestrian movement direction. Most existing datasets related to this topic, such as the Caltech [23] benchmark, the Pascal database [24] or the INRIA person dataset [25] focus only on pedestrian detection, providing bounding boxes of the detected pedestrians, but no information about the pedestrian direction or intention is provided. The Daimler dataset [16] is the only one that not only provides ground truth information for pedestrian detection tasks but also pedestrian intention information. Four different pedestrian motion types are considered: crossing, stopping, starting to walk and bending in. Each video sequence has the previously mentioned label about pedestrian intention. In our dataset, we considered different motion types, not previously included in existing datasets. We decided to recognize more generic motion types, such as moving left, right or to the front, which can be used to recognize higher-level motion types like those proposed in the Daimler dataset.

This dataset was manually annotated, creating a ground truth split for training the proposed CNN architecture. The dataset comprised 7416 images for training, 1752 for validation ad 1632 for test. Images were classified according to three different categories, right, left and front: 4035 images were assigned to *Right*, 4227 images to *Left* and 2538 to: *Front* (see Figure 2). Images within each category where randomly assigned to each of the three splits (training, validation and test).

The proposed dataset is available on our project website http://www.rovit.ua.es/dataset/pedirecog/. We carried out experiments to test different aspects of the proposed CNN architecture, such as layer configuration, solver values and train/validation/test dataset splits. Several hours of video were recorded in five different locations.

IV. Pedestrian direction recognition using CNNs

This section describes our method for pedestrian trajectory recognition. It is based on a CNN network trained with our own dataset and taking preprocessed data from a video feed as input. Once the CNN was defined, and the dataset was annotated with ground truth information, we started to feed the CNN with training data from the recorded dataset.

In the proposed method, acquired video is passed through an image preprocessing pipeline for image filtering, obtaining our final added video frames (Figure 3), which were used as input data for the proposed CNN architecture. Only one of every six frames was used, as we saw that consecutive frames at 30 frames per second contributed no new relevant features to our network.

First, we computed a dense optical flow using the algorithm proposed in [26], allowing us to detect changes in the scene or camera movements. Although some works have directly used optical flow and some simple heuristics to predict pedestrian movement direction, we observed that a dense global optical flow does not provide proper direction data due to movement of other objects or camera movements or a mix of both in the same scene, resulting in poor classification results.

In our pipeline, once we detect movement in a frame, we move on to the next stage, detecting pedestrians in the scene using an existing technique based on HOG features and a linear SVM ([1]). If we detect a pedestrian in the scene, we start to compute the optical flow of the image sequence (Figure 3, second row). Then, using two consecutive segmented images from the detected pedestrian region and the optical flow, we perform an image subtraction step, obtaining an absolute difference of two frames (Figure 3, third row). In this way, the contour of the pedestrian shows a slight movement in a particular direction, as seen in the lower part of the Figure 3. Then, we continue processing the sequence obtaining the next subtracted frame. Finally, using consecutive subtracted frames, we perform the sum (pixel level) of these subtracted frames obtaining a similar black background frame (Figure 3, fourth row). This output image is used as input data for the proposed CNN architecture, which automatically learns to extract features from these preprocessed images.

The images are then resized (downscaled) to 216x160 pixels. In addition, as the colour provides no further relevant information, we decided to convert the images to greyscale just after the acquisition step. Finally, we trained CNNs using modified versions of different existing CNN architectures such AlexNet [9], GoogLeNet [19] and ResNet [20], which were modified for the recognition of the proposed classes and fine-tuned for our specific problem.

One of the key elements for this work was the proposal of a novel input representation for the CNNs. An example can be seen in Figure 4, where the output frame produced as a result of the pre-processing steps highlights pedestrian motion from the rest of the scene.

V. Experiments

In this section, we describe the experimental setup and the different experiments carried out for the validation of the proposed method. Firstly, we have made an experiment using a computer vision based approach, showing a base line to compare. We evaluated the system using different Convolutional Neural Networks. In order to evaluate the performance of our proposal in terms of accuracy, we used the dataset presented in Section III. First, we tested the system using an AlexNet network and performed an exhaustive hyperparameter tuning process in order to boost the accuracy of the proposed system. Then, we performed data augmentation and evaluated it using AlexNet network. Finally, we tested the proposed pipeline using two of the best performer CNN networks for image recognition: GoogLeNet and ResNet. Finally, we present qualitative results showing the accuracy and the computational cost of the proposed system using different CNN architectures.

A. Methodology

All timings and results were obtained by conducting the experiments in the following setup: Intel Core i7-5820K with 32 GiB of memory. Additionally, the system included a NVIDIA GTX 1070 used for training and inference. The framework of choice was Caffe RC2 running on Ubuntu 16.04. To perform hyperparameter tuning we used the optimization tool *Spearmint*. This software tool uses *Caffe* and performs a Bayesian optimization based on a previously established range of parameters.

B. Computer vision-based approach

As a baseline to compare with the proposed approach, we have carried out an experiment where it has been used the dense optical flow algorithm described in [27] and the gradients of a Motion History Image (MHI) [28] to detect direction of motion. Given consecutive images where a pedestrian has been detected, the optical flow is calculated and used to estimate the direction of the segmented pedestrian. It was empirically tested using different thresholds for segmentation and MHI computation, choosing the ones providing best performance (MHI duration = 0.05 milliseconds, segmentation threshold = 35 (HSV distance), Max time delta = 125000.0 and min time delta = 5). Optical flow global direction is calculated as a mean weighted direction using all moving pixels. This provides us with a value between 0 and 360 degrees. Finally, those values were discretized in three ranges (120 bin size), for each predefined direction: left, right and front.

Table I shows quantitative results on the proposed dataset using the approach described above. It can be observed that this approach performs poorly in the test set, obtaining an average accuracy of 51%, 39% and 40% respectively for each motion type.

TABLE I
Confusion matrix results using optical flow and the gradients of a motion history image.

	Front	Left	Right
Front	0.51	0.35	0.14
Left	0.36	0.39	0.25
Right	0.38	0.220	0.40

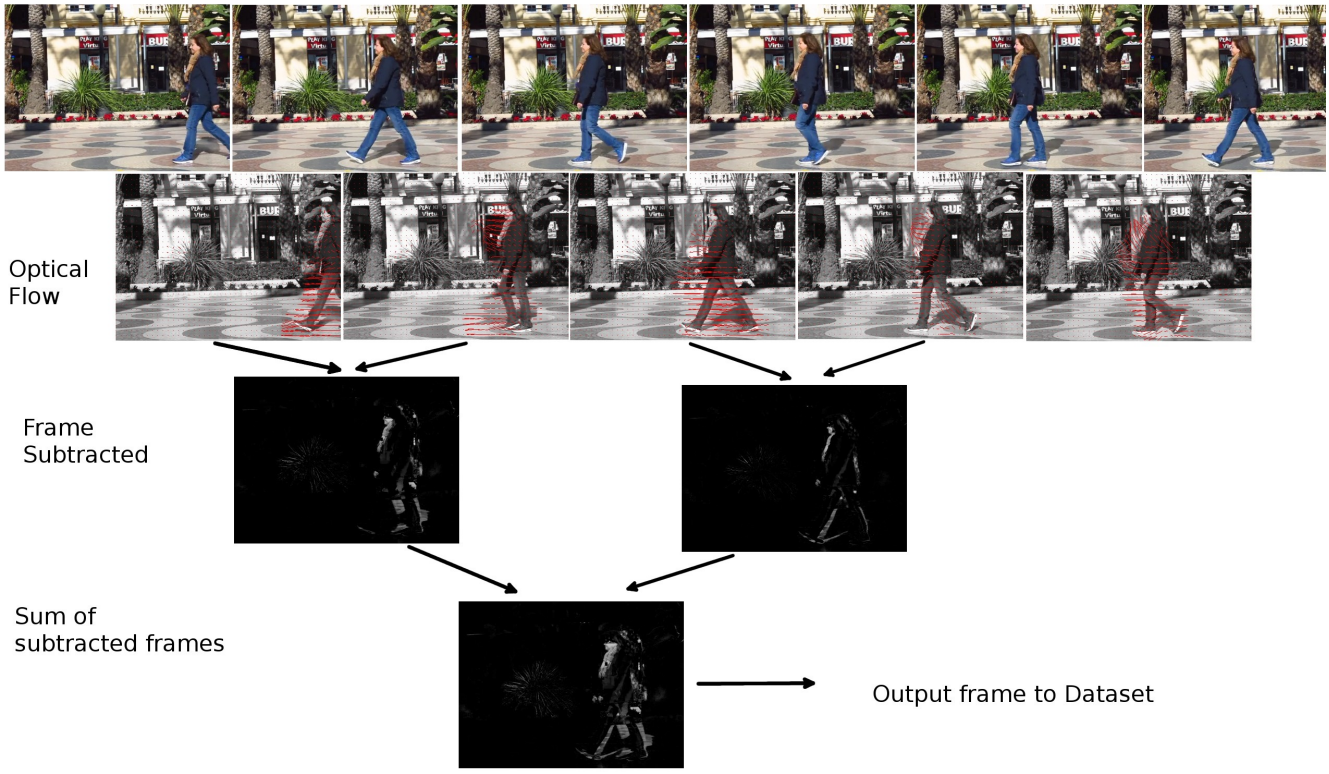

Fig. 3. Full process for video frame extraction of a pedestrian in motion

Fig. 4. From left to right: Sum of subtracted frames, detected pedestrian, Optical flow, a video showing an example of the preprocess input video stream can be found at: https://youtu.be/5dN4oePAqTg

C. AlexNet evaluation

For the evaluation of the pipeline using AlexNet, we trained the proposed CNN using Stochastic Gradient Descent (SGD) as solver type, with a batch size of 100 images. For the momentum parameter we tried values ranging from 0.8 to 0.95, and a weight decay strategy ranging from 0.0001 to 0.01. In most of the layers of this network, the weights were initialized using a Gaussian distribution with a standard deviation of 0.01. Then, the sixth and the seventh layers were initialized with a standard deviation of 0.05. The learning rate started as 0.01; we used an automatic reduction of 0.1 every 372 iterations, with a learning rate value of 0.0001 in the last iterations.

The layers structure is similar to the AlexNet [9] network. However, in this case, the input layer is a single image of 216x160 pixels (the image is converted to greyscale) and the last fully connected layer has only 3 neurons instead of 1000. No pre-trained model was used in the main experiment. Finally, two fully connected layers were configured with a 50% dropout to avoid over-fitting as was demonstrated by Hinton et al. [29].

Additionally, we ran a hyper-parameter search, in order to find the best values for the momentum and weight decay parameters. After this process, classification accuracy was improved for the proposed architecture, finding optimum parameter values for the CNN.

As mentioned, the same network model was trained using the same dataset, but in this case, the system was trained multiple times using a combination of several parameters. The momentum parameter was explored, considering values from 0.8 to 0.95, and also the weight decay parameter, exploring a set of values ranging from 0.0001 to 0.01.

1) Boosting accuracy: In order to further enhance the average 82% classification accuracy obtained using the AlexNet network, a series of test were conducted experimenting with several aspects of the whole training process, such as using Nesterovs Accelerated Gradient Descent [30] and a pre-trained CNN (fine tuning). Moreover, in order to provide our deep learning-based system with more ground truth information, we increased the dataset using data augmentation techniques. Finally, different batch sizes were evaluated.

We ran a few training steps using Nesterov's accelerated gradient solver and the same dataset previously employed. In all executions a 50% dropout was used, with no sample cropping and a batch size of 100 samples. A pre-trained model was used for the AlexNet network and for the first experiment we initialised network weights for the first convolutional layers, and the last fully-connected layers (6h and 8th layers). This change improved the accuracy of the network obtaining an 83% classification accuracy. For the second experiments we initialised the weights using Xavier's [31] method instead of the Gaussian method, and in the third one, a similar weight initialization was performed but only applying the changes to the sixth and final fully connected layers.

Finally, an additional test was carried out using a sharpened version of the recorded dataset (Figure 6). The whole dataset was filtered using a sharpen filter. The sharpen filter accentuates the edges of the input images. This proved to produce better results in terms of the test classification accuracy, but the loss in the validation data yielded worse results. Additionally, reducing the batch size without increasing the learning rate proved to be unsuccessful. According to Krizhevsky et al. [9], reducing the batch size by x, should be followed by increasing the learning rate by x. Some experiments tend to increase this by using the $\sqrt{x}$.

D. Results

The accuracy of the evaluated CNNs was quantified using the validation/test splits created during the dataset generation. After many different executions using hyper-parameters searching for the AlexNet CNN, we obtained several results ranging from a 74.8% to 83.8% classification accuracy. Best results were obtained for a value of 0.95 for the momentum parameter, and a value of 0.008127 for weight decay parameter. It can be seen that the loss function in all the cases also tends to a minimum of between 0.1 and 0.3. (See Figure 5 for an example of runs).

TABLE II
EXPERIMENT RESULTS USING NESTEROV'S ACCELERATED GRADIENT DESCENT SOLVER (ALEXNET)

Learning rate	Layers modified	Xavier init	Accuracy	Loss	120 test
0,01	no	no	80%	0,12	84%
0,01	Conv1, fc6, fc8	no	**82%**	0,14	85%
0,01	Conv1, fc6, fc8	Conv1, fc6, fc8	80%	0,09	86%
0,01	Conv1, fc6, fc8	Fc6, fc8	80%	0,04	85%

Testing Nesterov's solver, it was noticed that just initializing our own weights during the first, sixth and final layers, yielded better results than using just their pre-trained weights (see Figure 5).

TABLE III
RESULTS OBTAINED USING THE SHARPENED VERSION OF THE DATASET

Batch size	Layers modified	Xavier init	Accuracy	Loss	120 test
50	Conv1, fc6 fc8	Conv1, fc6 fc8	84%	0.05	64%
100	Conv1, fc6 fc8	Conv1, fc6 fc8	82%	0.47	83%
100	Conv1, fc6 fc8	Conv1, fc6 fc8	**83%**	0.53	84%

We can see in Figure 7 (left) that boosting the accuracy of the AlexNet network, we were able to achieve 84% accuracy. Additionally, based on the different experiments and results that were carried out while boosting AlexNet accuracy, we evaluated the proposed pipeline using GoogLeNet and ResNet-10. We omitted the results obtained using ResNet-50 since no improvement was achieved by using this deeper version of the ResNet network. Figure 7 shows accuracy and loss (training and validation) obtained using these three different CNN architectures.

GoogleNet and ResNet CNNs improved the results obtained using AlexNet. GoogLeNet achieved 86% accuracy and the best performer was ResNet achieving 94% accuracy on the validation set.

Finally, Table IV shows confusion matrices for each network using the test split. It can be observed that ResNet is again the best performer, achieving 79% accuracy on the test set. GoogLeNet and AlexNet achieved 77% and 71% accuracy, respectively.

TABLE IV
CONFUSION MATRIX FOR THE DIFFERENT ARCHITECTURES. FROM TOP TO BOTTOM: ALEXNET, GOOGLENET AND RESNET.

	Front	Left	Right
Front	0.920	0.042	0.036
Left	0.007	0.714	0.278
Right	0.012	0.374	0.612

	Front	Left	Right
Front	0.917	0.051	0.031
Left	0.011	0.772	0.215
Right	0	0.302	0.697

	Front	Left	Right
Front	0.980	0.011	0.008
Left	0.058	0.841	0.100
Right	0.081	0.265	0.652

We are also interested in the computational efficiency of the proposed system. Therefore, different experiments were carried out to measure the runtime for each stage of the proposed pipeline, evaluating the computational cost for each CNN architecture. Table V shows the inference time for each CNN architecture we evaluated. Table VI shows runtime for each stage of the proposed pipeline. The whole system has been implemented on the GPU using CUDA, achieving a framerate of 18 frames per second. Each frame takes around 53 ms to be processed from the moment the image is acquired.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a method to differentiate the motion of pedestrians in real life environments. By building a novel input-filtered image based on the post-processing of static

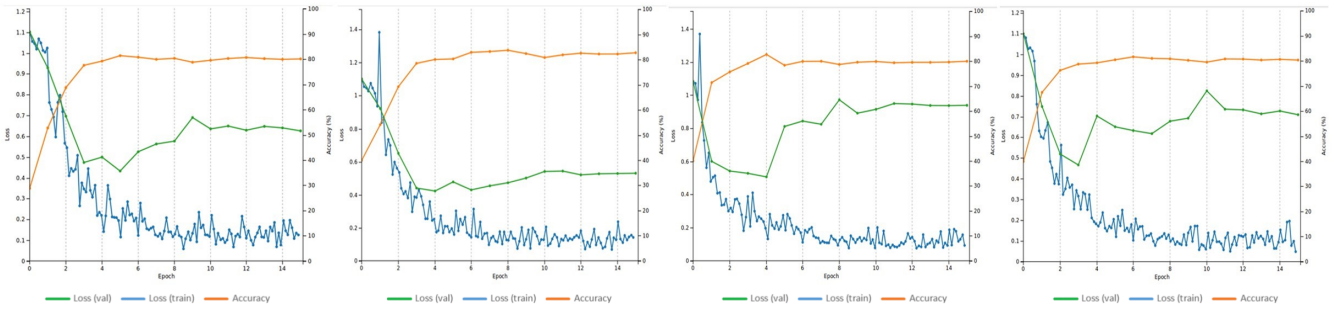

Fig. 5. From left to right: Nesterov's control model, first, second and third run. Layers. First, sixth and eight layers are not pre-trained and weights are not initialized in second and third run using Xavier's method.

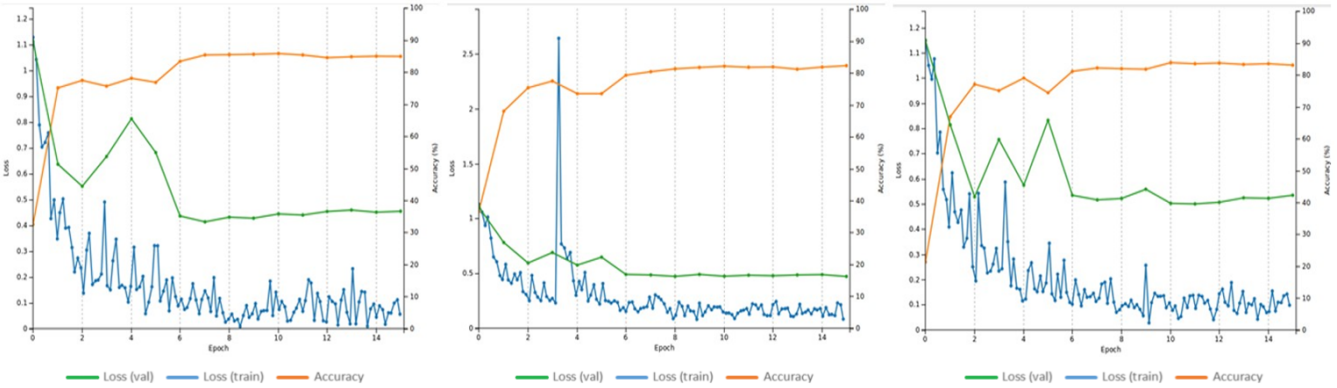

Fig. 6. Obtained results using data augmentation and the image sharpening step.

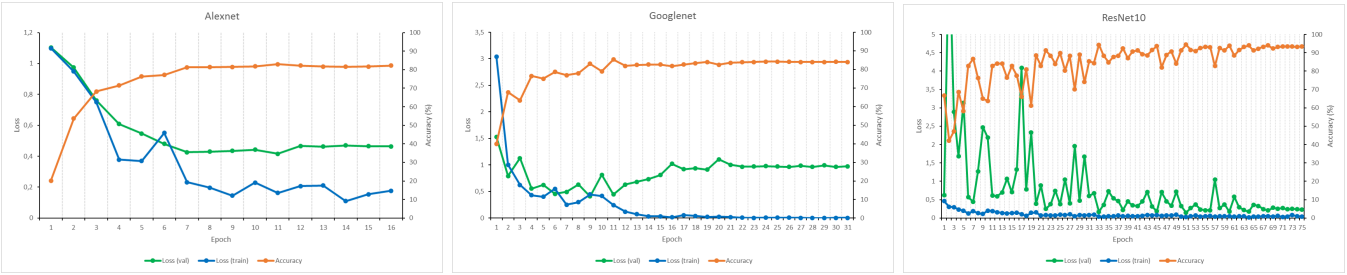

Fig. 7. Accuracy and loss (train and validation) for the three CNN architectures evaluated in this work.

TABLE V
RUNTIME FOR EACH CNN EVALUATED IN THIS WORK.

CNN architecture	Runtime (ms)
AlexNet	38.399
GoogLeNet	61.684
ResNet10	30.308

TABLE VI
RUNTIME FOR EACH STAGE OF THE PROPOSED PIPELINE.

Stage	Runtime (ms)
Image Preprocessing	1.512
People detection	8.341
Dense flow estimation	11.094
Sum of subtracted frames	2.132
ResNet	30.308
Total	53.387

recorded video frames, we have managed to successfully distinguish three different pedestrian movement directions.

Additionally, it has been proved how CNNs can impressively perform in such a task by training them with a specialised dataset. Moreover, we have demonstrated how the results can be enhanced even further by searching for the best hyper-parameters once the CNN has been fine-tuned for our specific problem, in this case tuning the momentum and weight decay CNN parameters. We have also presented an evaluation of the current state-of-the-art CNNs, with ResNet being the best-performing CNN for our image recognition problem (94% accuracy in the validation set and 79% in the test set).

As future directions, we are working on a better and more robust use of data augmentation, which should provide a more robust model. Online learning or incremental learning could also be beneficial, which would entail training the pre-trained CNN with new classified images from the same network. In addition, we are also planning to implement this pipeline in an embedded GPU platform, such as the NVIDIA Jetson TX2,

so we can deploy the developed system in a real ADAS.

ACKNOWLEDGMENT

This work has been partially funded by the Spanish Government TIN2016-76515-R grant for the COMBAHO project, supported with Feder funds. It has also been supported by the University of Alicante project GRE16-19. Experiments were made possible by a generous hardware donation from NVIDIA.

REFERENCES

[1] N. Dalal and B. 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*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[2] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection." in *CVPR*. IEEE Computer Society, 2010, pp. 1030–1037.

[3] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 99, 2009.

[4] R. Benenson, M. Omran, J. Hosang, and B. Schiele, *Ten Years of Pedestrian Detection, What Have We Learned?* Cham: Springer International Publishing, 2015, pp. 613–627. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16181-5_47

[5] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2011.155

[6] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, May 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb

[7] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[10] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] M. Enzweiler and D. M. Gavrila, "Integrated pedestrian classification and orientation estimation." in *CVPR*. IEEE Computer Society, 2010, pp. 982–989.

[12] T. Gandhi and M. M. Trivedi, "Image based estimation of pedestrian orientation for improving path prediction," in *2008 IEEE Intelligent Vehicles Symposium*, 2008.

[13] A. Mgelmose, M. M. Trivedi, and T. B. Moeslund, "Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations." in *Intelligent Vehicles Symposium*. IEEE, 2015, pp. 330–335. [Online]. Available: http://dblp.uni-trier.de/db/conf/ivs/ivs2015.html#MogelmoseTM15

[14] T. F. Fugger, B. C. Randles, A. C. Stein, W. C. Whiting, and B. Gallagher, "Analysis of pedestrian gait andperceptionreaction at signal-controlledcrosswalk intersections," *Transportation Research Record*, vol. 1, 2000.

[15] M. Goldhammer, A. Hubert, S. Köhler, K. Zindler, U. Brunsmann, K. Doll, and B. Sick, "Analysis on termination of pedestrians gait at urban intersections," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 1758 – 1763.

[16] N. Schneider and D. M. Gavrila, *Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 174–183. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40602-7_18

[17] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? a study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, April 2014.

[18] S. Koehler, M. Goldhammer, S. Bauer, S. Zecha, K. Doll, U. Brunsmann, and K. Dietmayer, "Stationary detection of the pedestrian?s intention at intersections," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 4, pp. 87–99, winter 2013.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: http://arxiv.org/abs/1409.4842

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[21] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013. [Online]. Available: http://arxiv.org/abs/1312.4400

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[23] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, June 2009.

[24] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, June 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0275-4

[25] 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 2005), 20-26 June 2005, San Diego, CA, USA*, 2005, pp. 886–893. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[26] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 25–36. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24673-2_3

[27] G. Farnebäck, *Two-Frame Motion Estimation Based on Polynomial Expansion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 363–370. [Online]. Available: http://dx.doi.org/10.1007/3-540-45103-X_50

[28] G. R. Bradski and J. Davis, "Motion segmentation and pose recognition with motion history gradients," in *Applications of Computer Vision, 2000, Fifth IEEE Workshop on*. IEEE, 2000, pp. 238–244.

[29] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[30] Y. Nesterov, "A method for solving a convex programming problem with rate of convergence $O(1/k^2)$," *Soviet Math. Doklady*, vol. 269, no. 3, pp. 543–547, 1983.

[31] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.

Alex Dominguez received a BSc in Computer Science from the University of Granada (Spain) in 1992 and a MSc in Artificial Intelligence from the University of Westminster (United Kingdom). He is currently a PhD student in the Department of Computer Science and Artificial Intelligence at the University of Alicante. His research interests include computer vision, deep learning, and autonomous driving assistance systems.

Miguel Cazorla received a BS degree in Computer Science from the University of Alicante (Spain) in 1995 and a PhD in Computer Science from the same University in 2000. He is currently full Professor with the Department of Computer Science and Artificial Intelligence of the University of Alicante. He has published more than one hundred papers on robotics and computer vision. His research interest areas are computer vision and mobile robotics (mainly using vision to implement robotics tasks). He has specialized on 3D computer visin, deep learning and human-robot interaction.

Sergio Orts-Escolano received a BSc, MSc and PhD in Computer Science from the University of Alicante (Spain) in 2008, 2010 and 2014 respectively. He is currently an assistant professor in the Department of Computer Science and Artificial Intelligence at the University of Alicante. Previously he was a researcher at Microsoft Research where he was one of the leading members of the Holoportation project (virtual 3D teleportation in real-time). His research interests include computer vision, 3D sensing, real-time computing, GPU computing, and deep learning. He has authored +50 publications in journals and top conferences like CVPR, SIGGRAPH, 3DV, BMVC, Neurocomputing, Neural Networks, Applied Soft Computing,etcetera. He is also member of European Networks like HiPEAC and Eucog.