On the use of Convolutional Neural Networks for Pedestrian Detection

Sergi Canyameres Masip

Abstract– In recent years, Deep Learning has emerged showing outstanding results for many different problems related to computer vision, machine learning and speech recognition. In this paper, we study the possibilities to apply convolutional neural networks (CNNs) and explore their power to address the pedestrian detection problem in the context of autonomous driving. We focus on creating a simple and robust framework based on the combination of a CNN architecture and a fast linear classifier. We show how the combination of these ingredients leads to a very accurate classifier, overcoming widespread techniques such as the HOG pedestrian detector and reaching state-of-the-art performance. Results from the wide range of experiments performed are analysed and compared on INRIA, one of the reference datasets for pedestrian detection.

Key words– Autonomous driving, pedestrian detection, deep learning, convolutional neural networks, domain adaptation, fine-tuning.

Abstract– En els darrers anys, el *deep learning* ha sorgit mostrant resultats excepcionals en diferents problemes relacionats amb la visió per computador, l'aprenentatge automàtic i el reconeixement de la parla. En aquest article estudiem les possibilitats d'aplicar xarxes neuronals artificials (*CNN* en anglès) i explorem el seu potencial envers la detecció de vianants en el context de la conducció autònoma de vehicles. Ens centrem en crear un marc de treball simple i robust, basat en la combinació d'una arquitectura de xarxa neuronal convolucional i d'un classificador lineal veloç. Demostrem com, amb la combinació d'aquests ingredients, obtenim un clasificador molt precís, superant tècniques tan exteses com el detector de vianants HOG, i arribant a un rendiment com el de l'estat de lart. Els resultats del gran ventall d'experiments fets s'analitzen i es comparen amb INRIA, un dels conjunts de dades més referents per a la detecció de vianants.

Paraules clau– Conducció autònoma, detecció de vianants, deep learning, xarxes neuronals convolucionals, adaptació de domini, fine-tuning.

1 INTRODUCTION

NEW techniques for Advanced Driving Assistance Systems and Autonomous Driving are progressing at a fast pace thanks to the lower cost of sensors and more efficient computing algorithms. However, truthful scene understanding is still the main challenge to implement intelligent applications such as collision prevention systems, lane trackers or pedestrian detectors. Whereas cars may be able to communicate between themselves in a future, and their movement is homogeneous and quite predictable, pedestrians can suddenly appear on the road because of their lack of vision field or a simple distractions. In fact, pedestrians represent around 6,300 annual deaths only in the EU.

Hence, robust Computer Vision solutions are crucial milestones if we aim to replace error-prone procedures by reliable systems that understand the surrounding scene with the visual information acquired by cameras.

Intelligent agents are capable of analysing their environment and learn from its characteristics for an specific goal, producing judgements that maximize the success of the decision made for that task. When trying to endow these agents with the power of vision, classic Computer Vision techniques have based their learning procedures on using hand-crafted feature descriptors such as HOG [1], LBP [2] or SIFT [3] to represent the scene. Some try to obtain holistic representations whereas others are part-based and use combinations of more specific descriptors to extract the features $\boldsymbol{x}$ from a given image. In any case, these architectures have proven to be good low-level representations when applied together with classifiers like Support Vector Machines [4], or Random Forests[5] as represented in Figure 1. These, learn a set of parameters $\boldsymbol{w}$ that optimally classify the given features $\mathbf{w}^T\mathbf{x}$. However, the final precision of these models is not enough for the critical tasks such as pedestrian detection and they fail to generalize well for changing environments, which has restrained their success in high-precision demanding applications.

To overcome these limitations, inspired by the human nervous system neural, networks provide a totally different approach to the problem of how we represent the world.

E-mail de contacte: scanyameres@gmail.com

Menció realitzada: Computació

Treball tutoritzat per: Germán Ros, Dr. David Vázquez i Dr. Antonio M. López Peña (Departament de Ciències de la Computació)

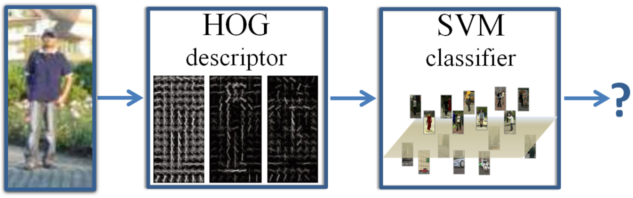

Figure 1: HOG extracts visual features **x** from an input image, which can be used to train a classifier.

Instead of basing its success on carefully hand-crafted functions, or visual, understandable patterns, these machine learning algorithms attempt to learn representations between nodes (perceptrons), which present a similar functionality to human neurons [6] as explained in Figure 2. By composing several layers with different types of connections and specific purposes, we are able to create bigger networks with higher expressiveness. A perceptron receives a series of inputs and performs a non-linear transformation whose result may reach a threshold and, as a consequence, activate the node outputs. When training with labelled data, we evaluate if the estimated output corresponds to the desired one and we provide feedback to the perceptron in the backpropagation step [7]. This balances the weights of the inputs or the threshold to adjust its performance so that it provides the expected output. If repeated enough times with the control of a small learning rate, the net will be reliable for new, unseen easy samples.

The quality leap with respect to the previous methodologies lies on the fact that not only the classification model $\boldsymbol{w}$ is learnt, but the complete object representation $\boldsymbol{x}$ fits the problem needs. With this, the descriptor produces features which are easier to classify by the support vectors machine, so the overall performance increases.

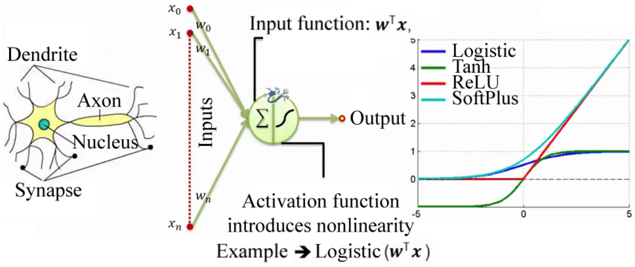

Figure 2: The activation function in a perceptron processes $\mathbf{w}^T\mathbf{x}$ to produce an output.

The original networks [10] were very shallow and therefore required an exponential amount of parameters to describe complex functionalities. In other words, when trying to approximate some complex functions with just one hidden layer between the input and the output, it requires an exponential number of parameters with respect to an equivalent multi-layer net (in terms of expressiveness). A more convenient strategy is to represent functions by combining the outputs of several layers, representing basic *blocks*, which drastically reduces the amount of parameters since these parameters are reused. This leads to the concept of deep network.

After years of research in artificial neural networks, we now understand better how to perform an effective training of deep nets. Together with the popularization of general purpose GPUs, that has made fast and large-scale trainings possible, the concept of learning is empowered to new horizons, and brings the possibility to explore deep learning and, as we do in this article, reach state-of-the-art detection rates by using these technologies. Among them, Convolutional Neural Networks are becoming very popular in the field of computer vision because of their similitude with the visual cortex structure. In the same way that our cells are sensitive to small sub-regions of the visual field, perceptrons in CNNs are arranged to act as overlapped local filters as Figure 3 represents.

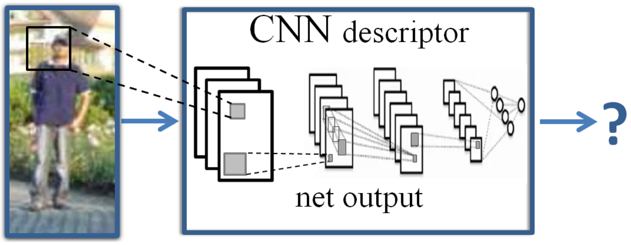

Figure 3: CNNs learn which convolution parameters **w** can produce better features **x** to easily predict an optimal output.

These deep representations have achieved state-of-the-art results in several computer vision tasks during the last years. For this reason is of great interest to study if they can be applied to the problem of pedestrian detection, and if this leads to noticeable benefits with respect to the current state-of-the-art based on DPM [8] or HOG-SVM [1] classifiers.

Our main goal is to *substitute the classical hand-crafted features for new ones learnt by general-purpose CNNs*, and to be able to apply them successfully to address the complicated problem of pedestrian detection. To this end we define a detection framework founded on the combination of a CNN based on the AlexNet [9] architecture, and a linear classifier, which we are going to use for the following tasks:

- Train an SVM image classifier with deep features extracted from the last fully-connected layers as shown in Figure 4 (Section 3.1).
- Add an intelligent candidate generator to improve the computational efficiency of the system (Section 3.2)
- In a following stage, study the influence of SVM bootstrapping and network fine-tuning, among other architecture alternatives (Sections 3.3 to 3.7).

In chapter 4 we perform a thorough analysis of the impact of the aforementioned elements on the final accuracy. By using the INRIA pedestrian dataset, properly fine-tuned models quickly outperform prevailing high-level systems such as HOGSVM.

2 State of the Art

Despite LeCun's first approach to convolutional neural networks in 1998 [10], to deal with document and digits recognition, these algorithms remained unpopular due to the slow

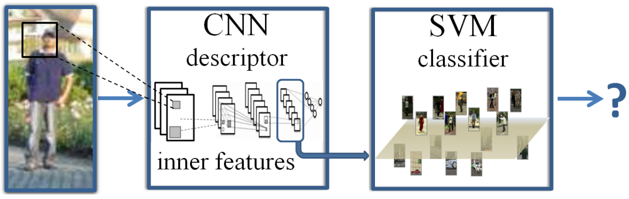

Figure 4: We propose to use the learnt features **x** from the last network layers to train a SVM classifier.

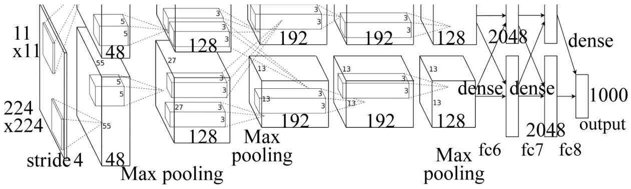

Figure 5: Alexnet's original structure: 5 convolution blocks (convolution + normalization), 2 fully-connected layers (fc6 & fc7) and a final fully-connected + softmax classifier.

and tedious training process required. It was not until Hinton's learning proposal in 2005 [11] that plenty of projects started to use them for multiple heterogeneous tasks.

Top results were reached in computer vision challenges such as pose estimation [12] feature matching [13], scene recognition [14], general object detection [15] and object classification [9]. Precisely, the impressive results from the latter projects in general object recognition challenges like the ILSVRC [16] attracted the attention of researchers dealing with pedestrian recognition. Problem-specific architectures started to appear [17] and showed solutions for dealing with small training data, a problem that we had to address in this article too. Other networks use cropped regions of different sizes from an image to obtain the contextual features and feed each layer, as MultiSDP, described in [18]. Later projects focused on complementing existing technologies failing in specific situations such as partial occlusions. For example DBN-Isol [19], which is based on a deformable part model combining Restricted Boltzmann Machines [20]. As far as we know, the current state-of-the-art applied to the problem of pedestrian detection is a Switchable Deep Network (SDN) by Luo et al. [21]. It uses different body parts to automatically learn hierarchical features and other mixture representations which allow them to properly separate background noise from the relevant regions. Despite of this solution, a recent implementation of low-level visual features and spatial pooling [22] slightly outperformed previous competitors in INRIA, ETH and Caltech-USA benchmark, showing that there is still much to analyse and discover in order to fully exploit the capabilities of CNNs. This work has been done under this premise.

For further information, comparative studies have been published regarding CNN implementation details [23][24] and analyzing existing pedestrian detection models [25].

3 Development

The version of AlexNet architecture [9] used all along this project surpassed all competitors in the ImageNet LSVRC 2010 and 2012. This challenge consisted on the classification of 1.2 million images in 1000 classes. The net (Figure 5) is based on five convolutional filters, which combine max-pooling and dropout intermediate layers. These lead to three fully-connected layers prior to the final softmax classifier that normalizes the calculated probabilities for each of the classes. Krizhevsky et al. introduced the dropout regularization method to reduce the over-fitting in the fully-connected layers [26]. Hidden neurons with probability 0.5 have their output set to zero to avoid them to influence in the back-propagation.

Only in the 17% of the tested images the network failed to place the correct label as one of the 5 more probable classes. However, in this paper we have to focus on the top-1 predictions, which means that only the most confident output generated is taken into consideration. In this case, the 62.5% of the tested images were properly labelled. This is the reference we are going to use to compare the evolution of our experiments against classical methods like HOG-SVM. To do so, we propose a series of modifications on AlexNet to achieve state-of-the-art results.

3.1 CNN features + SVM

Despite of AlexNet's impressive accuracy on general image recognition, a 1000-class classifier is far from optimal when the problem to face is reduced to pedestrian detection. The learnt classes do not even include high-level concepts such as person, man or woman, but much more detailed objects like jeans, tie or sandals. Furthermore, as seen in Figure 6, ILSVRC images are usually well-defined objects in clear backgrounds, without noise or occlusions, quite different from the street images acquired from a car. Therefore, applying the net off-the-shelf would fail to recognize pedestrians in complete frames crowded with different identifiable objects. Moreover, our goal is more complex than AlexNet's. Whereas it simply *classifies* a given image, we want to *detect* pedestrians, which implies not only classification but also location of the object within the frame. Hence, we need to adapt the procedure and add a candidate generation method. For example, our multi-size sliding window is applied across the image to produce patches of $H \times W$ pixels called crops, which are going to be used as working units. Our proposal is to train a Support Vector Machine classifier with the features extracted from these crops in the last layers of the net. For each j_{th} crop of the I_{th} image, C_I^j, we have a deep feature x_I^j extracted by AlexNet. Each of these features are extended with a ground truth label $y_I^j \in \{0, 1\}$ in order to train the SVM classifier.

Firstly, we take the AlexNet model, pretrained on ImageNet for a good feature generalisation, and we feed it with all the image regions that the sliding window produces from the INRIA training database. The network applies its trained filters and forwards the feature blob towards the final fully-connected layer and the softmax classifier. As we only want the deep features, we proceed by extracting the output before the softmax and the final dropout layers, i.e. fc6 and fc7, according to the definition of AlexNet. Instead of taking the probability estimation of the 1000 ImageNet classes, our idea is to use the 4096 output values after the fc7 and fc7 fully-connected operations, because they gained both general and specific information from all the convolu-

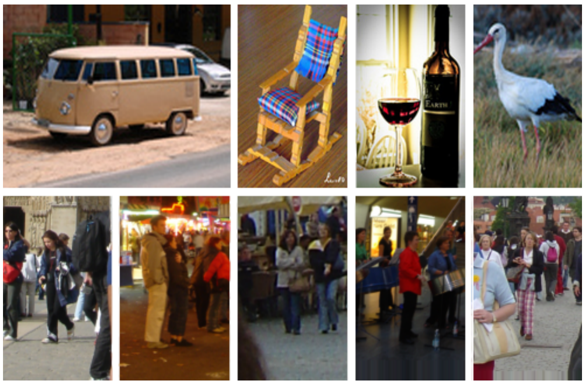

Figure 6: Images from the ILSVCR database (top) mostly contain simple objects with clear backgrounds. On the other hand, street images for pedestrian detection like from the INRIA dataset (bottom) are crowded and objects can be confused with the background.

tional filters, but at the same time they have not focused on the final classification yet. These features serve to train an SVM, with solely the two possible classes (pedestrian or not pedestrian). In testing time, the network extracts the features from the test images, which are subsequently given to the SVM models to calculate a confidence value corresponding to each crop. This pipeline is shown in Figure 5, the input image as a sub-region of a bigger frame cropped by the sliding window.

3.2 HOG Candidate Generator

Although the mentioned framework achieves good results, it is computationally very expensive. In the version of sliding window used, thousands of crops are generated for each original frame. Even if the evaluation time of the CNN when forwarding the region through the net is only about 20 ms, the entire dataset requires circa 45 hours to be completely processed. To avoid this, we propose to use a more sophisticated candidate generator based on a HOG-SVM classifier, adjusted to produce a very high recall, allowing only those images with a minimum chance of being a pedestrian to be processed. Setting a threshold value at -1 we can skip up to 98% of easy negatives relative to sky, buildings, empty road or clear background with little or no chances of the region to correspond to a pedestrian, and experiment with an architecture that needs not more than an hour to analyze the INRIA test dataset. Moreover, our focus here is on developing a robust model to properly deal with the complex decisions where classical engineered detectors still fail, which is not affected by the omission of these easy instances.

3.3 SVM bootstrapping

The INRIA dataset has a limited amount of positive samples, but we can use almost unlimited negative by choosing random areas of the images which do not contain any pedestrian. This is enough for training our SVM classifier acceptably well and leads to acceptable results. However, difference in quantity between the two classes samples may lead to a biased decision boundary, especially because many positives are in well-defined backgrounds, easy to separate by the SVM hyper plane. To address this issue and reduce the bias, we propose to apply a bootstrapping stage and train new SVM models. To this end, we test the net with only negative images, and save those with higher confidence estimations of being a pedestrian. These —hereafter referred as hard negatives— produce features points close to the positive cluster so that they are difficult to classify properly. By adding them to the training list of negative samples, the vectors separating the negative and the positive class will take these values into consideration, increasing the accuracy around a 5%.

3.4 Fine-Tuning

So far we have seen how well the original AlexNet architecture trained on ImageNet can perform in order to obtain useful deep features for pedestrian detection. If we wanted to train such a network for our specific problem, we would require a massive amount of properly labeled data samples in the order of a million images. However, we know that most of its inner features are universal enough to perform acceptably well with the appropriate classifier when processing natural images, which simplifies the training process by means of a simple domain adaptation. Hence, we do not need to train a full model, but adapt the existing one to our needs instead. Previous studies of domain adaptation for pedestrian detection have shown very good improvements when transforming a generic classifier into a problem-specific expert, even if introducing synthetic data to help the specialization [27]. Moreover, even if we cannot produce new labelled pedestrians, we can add some extra negatives by cropping multiple regions from negative frames.

Thus, what we propose to do is not a complete training of our own network, but just a reparametrization of the last layers, which are in charge of finding small particularities corresponding to each of the 1000 classes. In pedestrian detection we want to focus the power of these operations in finding the presence of humans. Therefore, it is necessary to modify the structure of the last fully connected operation to produce only two output classes, which are connected to the final softmax decision layer. This block is the only part needed to be trained from scratch, as the existing weights are not valid anymore if the connection structure of the net changes.

In order to feed the decision layers with the optimal information for their task, the two previous layers, fully-connected fc6 and fc7, are slightly modified. This time, instead of retraining them all from scratch, we take the weights learnt after the ImageNet training, and lightly modify them. As this process is slower than the full training of the last block, the layer learning rates are set to 1 and 2, which are values much smaller than the used in the new layers (10 and 20), because they have to learn faster. This allows the inner layers to be refined concurrently with the classification block and potentially improve the features calculated after the backpropagation correction. Our network is now specialized in detecting pedestrians, which greatly increases the accuracy up to an 85.62% if we use its deep features. Moreover, as we only have two classes, we may also use the net output directly to classify the images.

By always using the same INRIA positive and negative train images plus 20,000 random negative crops, this process takes around 8h to iterate 100.000 times with batches of 50 images in an NVidia GPU Tesla K40 boosted by the cuDNN library.

3.5 Dataset improving

A truly critical factor to make the fine-tuning process perform optimally is the size and variability of the data provided. As we can only use the INRIA training dataset, it is likely that the 60 million parameters in the network do not have enough instances to let the new architecture learn properly. However, and similarly to the bootstrapping process done for improving the SVM classifier, we can look for a big amount of hard negative crops and include them in the fine-tuning dataset in order to improve the deep features as well, instead of executing this process adding only poorly selected negatives, which can correspond to very easy regions such as free road or sky.

Our first fine-tuning experiments with only the INRIA dataset showed a remarkable increase on the network performance. However, after the fine-tuning with the explained extended dataset, the accuracy of the SVMs trained with the features of any of the last four layers clearly overcomes our state-of-the-art reference HOGSVM. All result tables are shown and explained in section 4.5.

3.6 Fine-tuned network without SVM

The alteration of the original AlexNet structure implies the possibility of directly using the class probability as the value of detection confidence, otherwise provided by our SVM. The last fully-connected (fc8) layer transforms the 4096 input features into two outputs which are normalized by a softmax classifier to produce both classes probabilities.

Alternatively we also fine-tuned the network with a Hinge layer instead, a loss-function typically used to train SVM classifiers. As the images still have a certain improvement range, it makes sense to think that the Hinge can learn from the potential of the features as the SVM does. With this, a better performance when classifying difficult instances is expected. Section 4.6 compares the performance of this alternative with respect to the previous procedures explained.

3.7 Other alternatives

Thanks to the recent rise of the use of convolutional neural networks there are still innumerable modifications that can possibly help know more about this technology. Here we expose three more experiments that we run with different degrees of success.

- fc6+fc7 combination: The efficacy of SVM classifier directly depends on the variability and number of data available. Nevertheless, too many instances can lead to overfitting, outliers can dramatically work against a potentially good set of features, and eventually there are not enough dimensions to properly fit an hyperplane in between. To face this, we propose to understand both fc6 and fc7 values as combined features of a same temporary instance. With this, we double the amount of good (hard negatives) training instances, but above all, we also double the number of features available so that the SVM can properly look for better plane combinations.

- Net architectures: since the very beginning we are focused on Krizhevsky's AlexNet due to its condition of 2012 ILSVRC winner and because the model works as reference baseline in open-source deep learning frameworks such as Caffe. However, other models have recently shown good results as well, such as GoogLeNet [28] or VGGNet with 16 or 19 layers [29].

- When fine-tuning, the defined learning rates control which layers are modified with respect to the baseline model and how much. The high-level convolutional layers generalize well for any images, so we can leave them untouched by setting their learning rates to zero. However, we want to retrain the fully-connected layers, so the learning rate will have increasing values as we approach the end of the network. Layers fc8 and prob, which experiment a full training from scratch, have the higher learning rates because they need to adapt faster.

  In section 4.7 we show the differences obtained when varying both these values and the sequence in which the layers are re-trained. Normally, architectures fine-tune all the last layers at once. We explored by incorporating a sequential re-training starting with only the last layer. Afterwards, the second-last layer starts re-training too, but with a smaller learning rate. Finally, we do the same with the third layer.

  A third version of this experiment consists of repeating this procedure, but setting the learning rate back to zero after each layer has trained, so that only one layer is fine-tuned at once.

4 EXPERIMENTAL EVALUATION

In this section we analyse the results of the different improvements explained along Section 3.

4.1 Analysing CNN features + SVM

Recognizing pedestrians from a camera in a car is not an easy task, especially when driving around crowded streets filled with infinity of different objects and backgrounds. For this reason, it makes sense that a CNN trained with easily identifiable objects is not suitable for distinguishing pedestrians in a chaotic scene, particularly if no specific training has been done. Hence, the accuracy levels shown in Table 1, by using the vanilla configuration explained in 3.1, look far from the 83% of accuracy achieved by AlexNet in the object recognition challenge [9]. Even so, the valuable information emerged from the experiments confirm our hypothesis.

We observe how the performance dramatically decreases as we train the SVM with features extracted from lowerer layers, which corresponds to the fact that after the first fully-connected layer (fc6), the network is increasingly problem-specific. This means that all the valuable information for our purpose is gained in the first filters through the convolution process.

We also look at the individual outputs produced by each SVM configurations trained with different parameters. To produce significant conclusions we report statistics with the minimum, the maximum and the average accuracy of these SVMs. In all the cases, using fc6 consistently produces the best results. Please note that we discarded the outcomes produced by the last layer after the softmax classifier due to its specialization in ImageNet and consequent poor performance for our problem.

4.2 Analysis of the Candidate Generator

This subsection analyses the results of incorporating the HOG candidate generator as presented in section 3.2. Unless otherwise indicated, the HOG detector used to generate the confidences for each region has a threshold set to -1. As the confidence range goes from -2 (not pedestrian) to 1 (pedestrian), we know that most regions with confidence above 0 correspond to a pedestrian. Hence, setting it to -1 embraces almost the totality of positive regions in the dataset.

Table 2 shows how reducing the amount of input regions not only shortens the testing time, but also -and most important- increases the accuracy of the system an average of 13.4% and 21.5% for layers fc6 and fc7 respectively. All our accuracy values correspond to the percentage of positive cropped regions detected as such by allowing one false positive per frame. This 75% obtained means that, given a full image with a single pedestrian in it, we mistakenly label one region as positive, whereas we correctly identify the pedestrian in 3 out of 4 regions where it is contained.

From this big jump we conclude that most of our wrong classifications are not bypassed pedestrians (false negatives) but somehow human-like regions which are understood by the CNN as people, i.e., hard negatives. Hence, our idea to perform image bootstrapping is totally suitable to advance towards more reliable models.

4.3 Analysis of SVM bootstrapping

Even though one could think that the SVM models would be near the saturation and would not be able to learn much more despite of the multiplication of the training data, the truth is that some of the tested configurations gained up to 5 points or more. In fact, after a first small test with 1000 extra images the correct detections increase several points. As seen in Figure 7, after the first jump, the remaining improvement is reached in an almost linear behavior when using 10k, and 30k new instances, until saturation starts being noticeable. If bootstrapping continues up to 50k extra images, not only the improvement gets stuck, but the overall performance also decreases due to the extreme over-fitting caused by the big amount of hard negatives provided in comparison to a now unnoticeable set of positive samples.

	Extraction layer		
	fc6	fc7	fc8
Minimum	56.33%	45.03%	31.68%
Maximum	**62.67%**	**49.14%**	**45.72%**
Average	58.80%	46.56%	37.80%

Table 1: Accuracies of the different SVM models trained with features extracted after the fully connected layers fc6, fc7 or fc8.

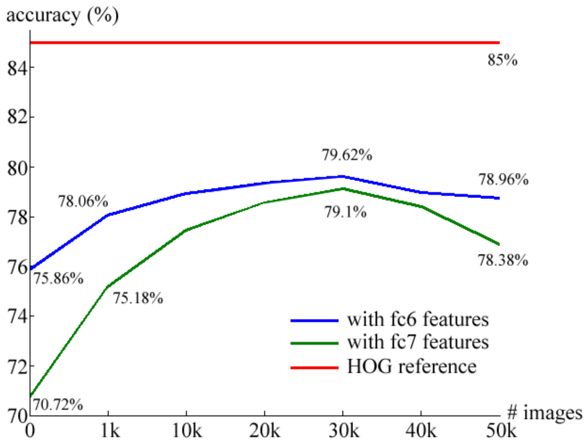

Figure 7: Effect of using different amounts of extra hard negative samples when bootstrapping the SVM models with features of layers fc6 and fc7.

As we can see, the best model produced has a 79.62% of accuracy, which is starting to be close to the 85% from the HOG implementation that we aim to reach after upcoming adjustments. However, this is the last experiment done using the original AlexNet architecture, which we modify to obtain a problem-specific CNN model that better fits our needs. For this reason, even though the results are good, these bootstrapped SVM models are left apart.

4.4 Analysis of the Fine-Tuning process

In Table 3 we show the results obtained after the fine-tunng process explained in Section 3.4. Despite the structural properties of the two first layers to remain untouched, they are capable to boost their relevancy an average of 9 and 15 points. Moreover, the variance among the different SVM parameters becomes nearly negligible, meaning that the maximums of 81.6% in fc6 and 84.07% in fc7 are not due to isolated cases of luck but because of the actual strength and robustness of the features produced by the network. Nevertheless, what is truly valuable after this fine-tuning process, is that the most meaningful layers are now fc8 and softmax, because of their total adaptation to our pedestrian detection goal. While the 1000-feature outputs could barely be used to get around a 60% of accuracy, the current 2-feature pre-

	Extraction layer			
	fc6	fc7	fc8	softmax
Minimum	69.86%	64.90%	57.36%	19.52%
Maximum	**75.86%**	**70.72%**	**63.87%**	**41.27%**
Average	72.23%	68.05%	60.95%	35.21%
HOG gain	+13.42%	+21.49%	+23.16%	+20.36%

Table 2: The accuracies of the SVM models trained with features after layers fc6, fc7 and fc8. If we filter the test images with a confidence lower than -1 after the HOG candidate generator, the results improve dramatically.

	Extraction layer			
	fc6	fc7	fc8	softmax
Max (all)	**75.86%**	**76.37%**	**81.51%**	**82.36%**
Avg (all)	75.04%	75.41%	79.63%	77.13%
Max (HOG)	**81.68%**	**84.08%**	**85.62%**	**84.93%**
Avg (HOG)	81.00%	83.42%	85.33%	83.30%

Table 3: The fine-tuned model is now specialized in pedestrian detection, so the last layers provide a big improvement with respect to the off-the-shelf network. Applying the HOG filter causes a smaller impact.

diction can properly guess more than 85% of the regions, which equals the HOG+SVM reference model.

These experiments prove that the inner layers are generic enough to face any kind of object recognition problem. Therefore, and similarly as in the case of bootstrapping, the key to keep boosting our CNN is going to be in the small details such as the learning rate values, or the size of the image batches used for fine-tuning. This matches as well with Chatfield et al. [24] studies.

Furthermore, we also checked the power of the fine-tuned model to perform without the HOG filter. In this occasion, the loss with respect to the filtered region test is much lower than when using AlexNet without fine-tuning. From the big differences seen previously in the last row of Table 2, we jump to a much more regular 5.7 - 8.2% loss shown in Table 3. This means that our intention to learn how to properly distinguish the hard negatives and reduce the ratio of false positives is accomplished. Please notice that in fact, all layers without the candidate generator produce, in average, better results (75.04%, 75.41%, 79.63%) than any combination of AlexNet even with the HOG activated filter (72.23%, 68.05%, 60.95%) as in Table 2.

4.5 Analysing the dataset improvement

In the same way that we improved the results when adding hard negatives in the bootstrapping process, the carefully selected images for this fine-tuning considerably outcome the detection rates obtained with the previous training set. If the original fine-tuning reduces the mismatch between the use of the different layers, repeating the operation with the new images nearly equals the performance of all layers to values reaching the border of an outstanding 90% of accuracy.

Again, the low variance within all the SVM models results is a sign of the robustness of the new model. Even in this small variation, no pattern relative to parameters and results can be inferred. Moreover, our problem-specific network (referred as *CNN**) has learnt to produce such reliable layers that the HOG candidate generator barely affects the final results. As we see in Table 4, the smaller difference occurs with the SVMs trained with the features after layer fc8. Removing the filter, this means, testing the totallity of the crops instead of just a 2% of the sliding window regions, implies a marginal decrease of 1.2% of accuracy. This difference was of 23.16 points on the original AlexNet.

	Extraction layer			
	fc6	fc7	fc8	softmax
Max (all)	**85.27%**	**86.99%**	**88.70%**	**87.33%**
Avg (all)	83.14%	85.98%	87.66%	85.20%
Max (HOG)	**87.67%**	**89.90%**	**89.90%**	**90.07%**
Avg (HOG)	87.67%	89.51%	89.90%	88.13%

Table 4: Fine-tuning with our improved dataset is the ultimate deep features boost. This *CNN** network becomes very robust and is better at detecting hard negatives, so the HOG filter has almost no effect on the final results.

4.6 Results of a fine-tuned net without SVM

The first attempt to directly use the estimated confidence value to contrast our model against HOG results without any classifier gave irregular results. Even if our baseline model produces much better results with the SVM than without it, Table 5 shows a big difference between the use of the default Softmax layer and the implementation of a new Hinge loss-function. However, these differences are dramatically reduced when applying the tests to a very similar model, fine-tuned with batches of 200 images instead of 50. Although its performance when using the SVM is almost the same, the loss without it is much smaller, and the Hinge classifier can even outperform the results obtained if we look at the confidences estimated at layer fc8.

Nevertheless, all these variances vanish when using the models fine-tuned with our boosted dataset. As we can appreciate, the Hinge version either underperforms or equals the Softmax models, because the confidences produced are good enough not to require an SVM to understand them. In fact, if we look at the consequences of removing the SVM after the Softmax process, the differences are reduced to the point that, for the 50-batch model, the raw confidences right after the fc8 layer produce an accuracy of 90.23%, which is the best result achieved in this project.

	Extraction layer			
	fc8	output	fc8(*)	output(*)
(50) SoftSVM	85.27%	85.78%	89.89%	**90.07%**
(50) Softmax	74.41%	72.94%	**90.23%**	87.15%
(50) Hinge	82.02%	78.42%	84.93%	82.36%
(200) SoftSVM	84.24%	85.44%	89.04%	89.21%
(200) Softmax	83.72%	80.47%	88.52%	86.81%
(200) Hinge	86.13%	84.93%	88.01%	86.64%

Table 5: Results of different architectures combining Softmax or Hinge classification layers, the use of SVM, and batch sizes of 50 and 200. (*) corresponds to CNN models trained with the improved dataset.

4.7 Analysing other alternatives

- fc6+fc7: The accuracy for all SVMs after layers fc6 and fc7 are 72.23% and 68.04% respectively, so together they form an average of 70.14%. After appending the features from both layers to form single instance, the accuracy obtained is 71.95%. Even if it

	Extraction layer	
	fc8	output
Softmax	**90.23%**	**87.15%**
Holding 2 it.	85.79%	79.62%
Sequential 2 it.	86.30%	83.90%
Holding 3 it.	87.67%	83.39%
Sequential 3 it.	85.45%	83.04%

Table 6: Comparison of our non-SVM models. *Holding* and *Sequential* fine-tuning variants for two and three iterations underperform the standard procedure.

gains 1.81% with respect to the two independent classifiers together, we cannot consider it as a significant improvement as it is still below the result after using the fc6 alone.

- Other models: the results obtained were not satisfactory. Unfortunately, GoogleNet is an ensemble of current models, and the versions of VGGNet extend the depth of the network up to 16 and 19 layers. This causes both architectures to run 3x to 13x times slower and requires an intense study from their structures which is not in the scope of our project.

- So far we have seen how the network can learn when fine-tuning all layers at once, gradually increasing values of learning rates. Table 6 shows the results obtained if performing the fine-tuning with the strategies explained in section 3.7. *Holding* corresponds to the first variation described where all we gradually activate the learning rates, and *Sequential* corresponds to individually fine-tune the layers one after the other in successive iterations. As we see, both subtly underperform the all-at-once fine-tune results obtained so far. However, this is only a first approach and infinity of other learning rate combinations and sequences can be attempted in the future, so more research can still be done in this direction.

4.8 Discussion of the results

In this article we have shown how the use of deep features learnt by a general-purpose CNN as an alternative to older pedestrian detection methods is totally feasible. Achieving the initial goals, we have been able to reach state-of-the-art results.

- The use of the extracted features to train an SVM has led to good results all along the experiments. We could learn more about the effects of the layers and how to use the features from different depths depending on the specific modification tested, with accuracies varying up to 20 points in the off-the-shelf AlexNet model.

- Adding a candidate proposal method allowed us to have more flexibility and speed when testing different implementations. Moreover, thanks to it we could also understand better the evolution of the results, which were improved between 13.4 and 23.2 percentual points by more reliable models with a 40x speedup.

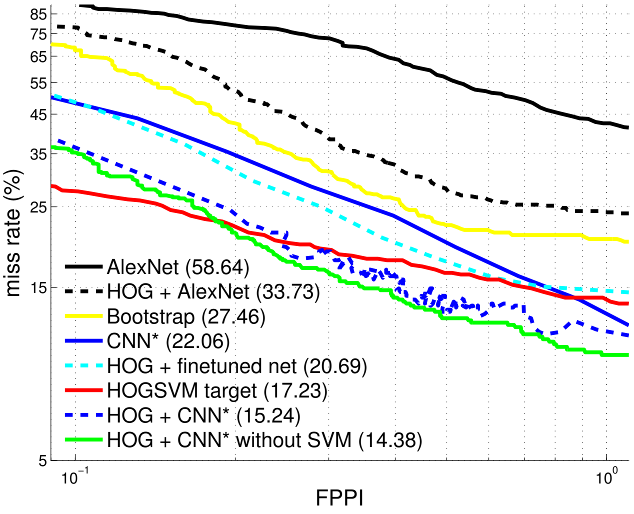

Figure 8: Average miss-rate depending on the amount of False Positives Per Image along the different architectures tested in this project. After fine-tuning the network with the improved dataset, our two best models outperform the state-of-the-art HOG+SVM model in almost 2 and 3 points, respectively. CNN* corresponds to the network fine-tuned with our improved dataset.

- The different boosting processes were successful enough to definetely let the results reach the state of the art. Whereas bootstrapping increased the accuracy up to a 7%, domain adaptation methods could add some extra 15% (fine-tuning) and extra 8% (dataset improving), achieving astonishing accuracies around the 90% even with the non-filtered detector.

In Figure 8 we show the full progress of all our experiments. This chart allows us to see not only the accuracies used for our benchmarks so far, corresponding to 10^0 False Positives Per Image, but also the miss rate increase when we move the threshold to allow less false positives. Even though the HOG+SVM decrease is smoother, the overall area under the curve is smaller in our architectures fine-tuned with the improved dataset. The features extracted from the softmax layer can train an SVM with 15.24% average miss rate, 2 points better than our HOG target. Otherwise, taking the detection confidences directly from the output of the network, we obtain the best result, with an average miss rate of 14.38% which improves in almost 3 points our target.

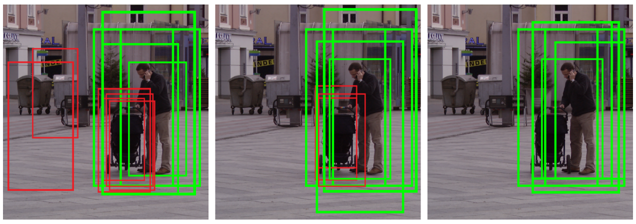
AlexNet + SVM HOG + AlexNet + SVM HOG + CNN*

Figure 9: Detection examples to show how our new architectures are more robust. False positives are dramatically reduced and the overall accuracy overcomes classical state-of-the-art models for pedestrian detection.

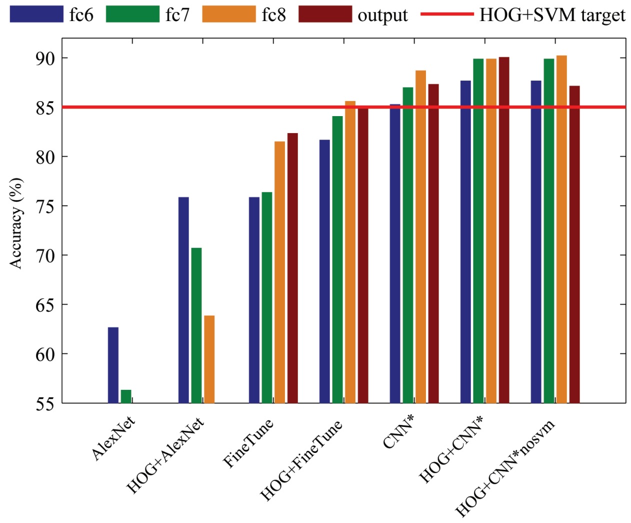

Figure 10: Best performance of the deep features extracted from the last four layers (fc6, fc7, fc8 and net's output) along the different architectures developed.

The improvement achieved along the project is better represented in Figure 9. The the classifier with original AlexNet's features produces multiple false detections, some of them in areas which apparently should not be confusing. Next, the HOG candidate generator drastically reduces the amount of easy false positives, even some messy regions are not filtered and can still be understood as pedestrians by the classifier. Finally, after all the boosting methods, very few false positives or false negatives are produced. The features are perfectly trained to properly detect most pedestrians, and the system is very robust against complex detections with crowded scenes, occlusions and other difficulties. As we can see in Figure 10, the performance of the deep features from the used layers balances and improves all along the multiple model modifications done.

5 Conclusions

Dealing with CNN is still a crucial task in new research projects, but lots of new publications on the topic are continuously appearing so the principles of this technologies are in permanent evolution and transformation. However, this project accomplishes the goals presented, and this paper has shown how convolutional neural networks can be boosted for pedestrian detection.

We have studied the use of a pedestrian detector by combining a candidate object proposal, deep features and a simple classification tool, with the idea to improve the overall system accuracy. Furthermore, this article highlights how classical techniques like bootstrapping can dramatically improve the performance of a convolutional neural network. Domain adaptation processes such as fine-tuning are also crucial to better adapt the deep features as seen along our experiments. With all this improvements together, our proposal achieves state-of-the-art results for pedestrian detection and overcomes a widespread method like HOG+SVM.

Moreover, we are happy to see that all the achievements, problems, solutions, results and conclusions produced along these months totally correlate with those revealed by up-to-date publications such as CVPR's *Taking a Deeper Look at Pedestrians* [15], or ECCV's *Strengthening the Effectiveness of Pedestrian Detection with Spatially Pooled Features* [22] and *Analyzing the Performance of Multilayer Neural Networks for Object Recognition* [30].

6 Future Work

This project opens the gate to plenty of new possibilities regarding the improvement of convolutional neural networks. The experienced acquired allows our group to start several investigation lines involving semantic segmentation and pedestrian detection.

The next goal is not only to recognize pedestrians within given crops, but to find and identify them in a region within a whole captured frame. For this, we need to perform a structural redefinition of the net architecture, much more exhaustive than a simple fine-tuning. We propose to deeply modify the conception of the features by changing the inputs from raw RGB values to codified strings of image representations where activated value in an would represent the presence of a pedestrian in that region. The main issue to be solved is the amount of data needed to be fully trained. For this, we believe that domain adaptation processes can help CNNs to improve their performance in classification and also detection problems.

What is clear is that convolutional neural networks have a high potential in computer vision. Hopefully, the scientific community will continue investing in their research and harness them to keep advancing to a future with plenty of intelligent systems that bring more safety and comfort to dangerous or tedious human activities.

Aknowledgements

We thank the support of the DGT project SPIP2014-01352. Our research is also kindly supported by NVIDIA Corporation in the form of different GPU hardware.

I would also like to thank Dr. Antonio López and Dr. David Vázquez for their interest and confidence in me and to let me take part in such a promising project at the Computer Vision Center. Moreover, this work would not have been so successful without the priceless knowledge and the patience of Germán Ros, who spent countless hours helping me in the worst moments. My sincere thanks to all of them for their support.

References

[1] N. Dalal, B. Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR, 2005.

[2] X. Wang, T. X. Han, S. Yan. An HOG-LBP Human Detector with Partial Occlusion Handling. In ICCV, 2009.

[3] D. G. Lowe. "Distinctive image features from scale-invariant keypoints," Intl. Journal of CV, 2004.

[4] C. Cortes, V. Vapnik. Support-vector networks. In Machine Learning journal, 1995.

[5] L. Breiman. Random Forests. In Machine Learning journal, 2001.

[6] F. Rosenblatt. The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. 1958.

[7] P.J. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. 1975.

[8] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. 2010.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998.

[11] G.E. Hinton, S. Osindero, Y. Teh. A fast learning algorithm for deep belief nets. In Neural Computation, 2006.

[12] X. Chen and A. Yuille. Articulated pose estimation with image-dependent preference on pairwise relations. In NIPS, 2014.

[13] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to sift. In arXiv, 2014.

[14] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In ECCV, 2014.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In arXiv, 2014.

[16] 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. In arXiv, 2014.

[17] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In CVPR, 2013.

[18] X. Zeng, W. Ouyang, and X. Wang. Multi-stage contextual deep learning for pedestrian detection. In ICCV, 2013.

[19] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. In CVPR, 2012.

[20] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. 2010.

[21] P. Luo, Y. Tian, X. Wang, and X. Tang. Switchable deep network for pedestrian detection. In CVPR, 2014.

[22] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In ECCV, 2014.

[23] H. Azizpour, A. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In arXiv, 2014.

[24] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In BMVC, 2014.

[25] J. Hosang, M. Omran, R. Benenson, B. Schiele. Taking a Deeper Look at Pedestrians. In CVPR, 2015.

[26] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, 2012.

[27] J. Xu, S. Ramos, D. Vázquez, A. M. López. Domain adaptation of deformable part-based models. 2014.

[28] Long, E. Shelhamer, T. Darrell. Fully Convolutional Models for Semantic Segmentation. In CVPR, 2015.

[29] K. Simonyan, A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In arXiv:1409.1556.

[30] P. Agrawal, R. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In ECCV, 2014.