UNIVERSITAT POLITÈCNICA DE CATALUNYA

Doctoral Programme:

AUTOMÀTICA, ROBÒTICA I VISIÓ

Research Plan:

Object Detection for Autonomous Driving Using Deep Learning

Victor Vaquero Gomez

Advisors:

Alberto Sanfeliu Cortes, Prof.

Francesc Moreno Noguer, Dr.

December 2015

Contents

1 Introduction & Motivation

Autonomous Vehicles (AVs) are increasingly gaining attention worldwide. Potential of this technology is clear and it is predicted that will dramatically change transportation as we know it today. Autonomous Driving benefits go from reducing contamination in urban areas by improving driving and fuel efficiency to help controlling the traffic flow and parking problems. In addition, autonomous vehicles will speed up people and freight transportation, as well as increase the security by reducing the human error.

There has been a strong uprising tendency in the autonomous driving technologies since in 2004 DARPA introduced its first grand challenge. It defied companies and research institutions to built and run their Autonomous Vehicles to complete a 150 miles circuit, but in that occasion none of the teams managed to finish the route, completing the best AV only around 7 miles of the whole trail. Nevertheless, just one year later up to 5 AVs were able to travel the full path. In 2007 a new Urban Challenge was proposed, composed of realistic every-day-driving scenarios with traffic rules, blocked routes, fixed and moving obstacles, etc. and in this occasion six teams managed to finish it. Just observing this fast evolution, we can make an idea of the interest that autonomous driving is raising.

In the last years the limits of autonomous driving have been pushed due to the fact that industry, governments and research institutions are investing vast amounts of both human-time and money. Fully autonomous vehicles have became a reality, as for example by November 2015, Google's self-driving cars had driven more than 1.2 million miles (approximately 90 years of human driving experience).

Technology transfer has also been produced, and research advances are being gradually introduced in current commercial vehicles in the way of Advanced Driver Assistance Systems (ADAS). Moreover, several companies (Nissan, Mercedes, Volvo, Tesla, etc) have announced their intentions to have, by 2020, different models of commercially viable autonomous vehicles [1]. From their side, governments are starting to legislate and nowadays some are allowing to test and *drive* AVs on their roads (i.e. California since 2012).

1.1 Motivation

Although Autonomous Driving are getting more and more important, this technology is still far from being mature. Our cities and roads are very unpredictable dynamic environments where multiple actors such as pedestrians, animals, street furniture or other vehicles coexist together. In this way, it is needed to provide Autonomous Vehicles with robust perception systems in order to correctly understand the environment, and therefore being able to interpret what is happening in the surroundings to act in consequence.

The primary way for an AV to get information from the environment is through a variety of on-board sensors. Nowadays, new affordable sensors such as automotive radars, multiple-layer laser-scanners or appearance and depth capturing cameras are coming to market and being incorporated in mobile robots and vehicles. As a consequence, new sources of information are available to be used in their perception systems.

Regarding the software development, in the last years significant advances have been done in the field of Computer Vision, Machine Learning and Mobile Robotics research. Remarkably are Deep Learning and specifically Convolutional Neural Nsetworks (CNNs) advances, which are showing promising results. CNNs have supposed an important breakthrough, and its results are beating the state of the art solutions on problems such as object detection and classification or semantic segmentation and understanding (even surpassing the human capabilities [2]). Either being used as a new way for extracting robust features replacing hand-crafted ones, or as an end-to-end trainable system, CNNs are of special interest nowadays.

This context motivates us to participate in the research and development of new and robust Convolutional Neural Network-based algorithms that will perform a key role in the perception systems of autonomous vehicles, substituting standard computer vision approaches.

Additional motivation for this thesis comes from the European FP7 *Cargo Ants* project, which in cooperation with Volvo Trucks (as one of the industrial partners) aims to develop autonomous trucks that can co-operate in shared workspaces for efficient and safe freight transportation in main ports and freight terminals. In this way, we aim to test and integrate the developed advances into these trucks.

2 Objectives

The main objective of this doctoral thesis is to exploit Convolutional Neural Networks capabilities to efficiently perform scene understanding in the autonomous driving context. We aim to achieve this high level objective by tackling and combining low level CNN-based features such as optical flow or per-pixel semantic segmentation. In this way, main objectives in our research are:

- On one hand, we have as objective the extraction of advanced and robust low level geometric features by using Convolutional Neural Networks. We will primarily focus on obtaining robust CNN-based optical flow estimations of a scene, as well as per-pixel semantic segmentation. These two low level features will allow us to get dynamic and class knowledge of the observed environment.

  At this level, we will also work on improving performance and generalization of previously proposed networks by combining synthetic and real data, as well as by performing domain adaptation with the introduction of specific application knowledge into the network.

  Other objective at this low level is to explore methods for performing multimodal feature integration. In this way, we aim to develop CNN networks that will be able to integrate the obtained low level features with information from other sensors (such as from RGB images or sparse laser depth measures).

- On the other hand, we plan to accomplish higher level objectives by exploiting the obtained low level features. In such manner, we will propose new CNN architectures and strategies for performing scene understanding. Additionally, we will explore the best way of introducing specific application knowledge with the intention of making our approach efficient for autonomous driving.

2.1 Expected Contributions

To accomplish the aforementioned objectives, we will firstly review the state of the art and provide a comprehensive view of Convolutional Neural Networks.

We will research and develop new CNNs for solving the problems of optical flow estimation, semantic segmentation and object detection. We will go deeply on the CNN capacities to integrate different sources of low-level features for performing high-level interpretation, so subsequently being able to achieve scene understanding. With this, we expect to provide deeper knowledge about CNNs as, from a scientific point of view there is still no much knowledge about its internal behaviour, which sometimes leads the development of better models to a trial and error heuristic approach.

Furthermore we will generate synthetic but realistic new datasets of urban driving scenarios, in cooperation with other research groups, to be used for training CNNs. With this new amount of information, we expect to contribute on studying the generalization capacities and performance of CNNs when trained only with synthetic data, as well as its domain adaptation when specific application knowledge is giving.

During this doctoral research, MATLAB® will be used along with the MatConvNet [3] toolbox making use of GPU computation. We expect to also contribute on the development of new CNN layers and characteristics for this framework, making them available for the community.

Scientific publications in international journals and international congress communications are expected to be derived from this doctoral thesis. Some of the work has already been published and can be found in Section 5.1.

3 State of the Art

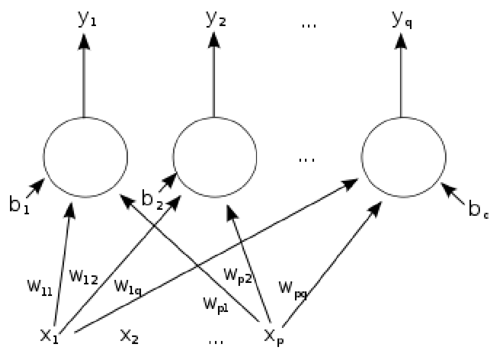
In this section we review the state of the art about the different topics related with this doctoral thesis. Firstly, we will focus our problem by doing a very brief review on classical feature-based object detection approaches used in computer vision and pattern recognition. Next, we will show the potential of Deep Learning techniques and Deep Neural Networks, which are performing stunningly well at solving computer vision problems when conventional approaches have typically had troubles. Specifically, we will focus our attention in Convolutional Neural Networks (CNNs) applied for feature extraction (such us optical flow estimation and per-pixel semantic segmentation) to solve problems as image classification and object detection.

3.1 Classical Object Detection in Computer Vision

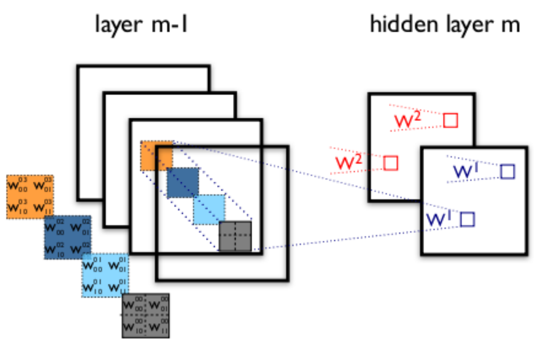
Standard methods for classifying or detecting objects in Computer Vision are commonly based in feature matching, which aims to find sets of pre-learned shape features (descriptors) of the objects across images. Some descriptors such as SIFT [4] or SURF [5] have been typically used at low level for locating points of special interest between images [6], [7]. Higher level features can be extracted with other advanced descriptors like Haar features [8] or Histograms of Oriented Gradients (HOG) [9], which have shown good performance in human and pedestrian detection [10], [11], [12] (an important application for autonomous driving situations). Best descriptors should be scale, rotation and illumination invariant as well as pose and occlusion aware. This usually involves hand-crafting and personalising them to the application or image space being used, which reduce its generalization capabilities.

Neural Networks [13] in pattern recognition have historically performed pretty well in situations where conventional feature-based approaches struggle. They were taken with great optimism in the 1980s, specially after backpropagation was expanded becoming widely known. In a standard Neural Network any neuron is connected to every other neuron in the adjacent layers, as can be seen in Figure 1(a). If more neurons are added, the connections will grow exponentially and the network may suffer from what is called the *curse of dimensionality*, which supposed a big problem back in the 80s. The output y_q of each hidden neuron is computed as the summation of each of the weighted inputs $w_{iq} \cdot x_i$ plus a bias b_q following the equation: $y = \sum_i w_{iq} \cdot x_i + b$. This means that a different weight and bias value must be computed for each neuron, and the total number grows as the network gets deeper. In example, for a 28x28 input image, a Neural Network would have 28x28 = 784 input units, and therefore for each of the hidden neurons on the first hidden layer, 784 weights and 784 biases values (1568) would need to be computed.

However, due to these dimensionality problems and hardware computing shortcomings, Neural Networks popularity felt down in the 1990s. Other machine learning techniques emerge taking Neural Networks popularity, such us Support Vector Machines (SVMs), Boosting methods and more recently Deformable Part-based Models (DPMs). Combination of HOG or Haar



(a) Detail of a Classical Neural Network



(b) Detail of a Convolutional Neural Network

Figure 1: Differences on the connectivity of standard Neural Networks and Convolutional ones. CNNs exploit spatial structure of the images by enforcing a local connectivity pattern between units of adjacent layers. In this way, units from layer m are only activated from a transversal region (through all its features maps) of units of layer $m-1$. Moreover, each feature map on a hidden layer shares the same weights and bias in all its units, so that they can detect the same kind of features. Thus, the number of learned parameters in comparison with standard Neural Networks is reduced, allowing faster training and deeper structures.

features with these methods worked pretty well in Object Detection and classification problems [14], [15], [16] and remained as the state of the art until some years ago.

3.2 Convolutional Neural Networks

In the last years, neural networks have again raised the attention of the research community. More specifically, Deep Neural Networks are setting all sort of records and defeating other approaches on many pattern recognition and computer vision problems. The success of these systems comes from their ability to create powerful object representations without the need to hand design features. Deep Neural Networks are able to learn both low level image features and high level abstracted concepts as well as abstract input-output relations given enough training data samples.

A Special family of Neural Networks is Convolutional Neural Networks (CNNs), which are enjoying nowadays of good popularity. As known today, CNNs come from the late 1970s, and were heavily used for different detection and classification purposes in the late 1980s and 1990s [17], [18].

3.2.1 Convolutional Neural Network Distinctive Concepts

The main advantage of CNNs over standard fully connected Neural Networks is that, inherently, they are able to take into account the local spatial structure of the images. Therefore, they do not treat far away input image pixels in the same way as those which are close by. Moreover, CNNs use a special architecture which is particularly well adapted to classification problems, allowing a faster training and therefore enabling the creation of deeper networks with many layers. Due to these special characteristics, deep convolutional networks and some close variant are used nowadays in most algorithms for image recognition.

Convolutional Neural Networks, in the same way as classical Neural Networks are composed of multiple blocks or layers stacked one after the other in such a way that outputs from one layer serve as inputs for the following one. In this manner, many of the standard Neural Networks concepts are applied to CNNs, such as backpropagation, gradient descent, regularization, etc.

However, CNNs allow us to train deep, many-layers networks by introducing three new ideas in order to avoid the dimensionality problem, to be: local receptive fields, shared weights, and pooling.

Local Receptive Fields.

For easily understanding how CNNs work over images it is common to represent its input layer as $height \times width$ arranged units as seen in Figure 1(b), which is different from the vectorial representation of classical Neural Networks. Convolutional layers input pixels are connected to the next layer of hidden units through the convolution process, which applies only over a region of the input map. This means that, instead of making fully connections between all the neurons, convolutional layers only make connections in a small, localized region of the input image.

Such localized architecture ensures that the learnt *filters* produce the strongest response to a spatially local input pattern. Although these filters apply locally, using many hidden layers leads to non-linear filters that become increasingly global, as they respond to larger regions of previous layers. Therefore, Convolutional Neural Network are able to create at lower layers simple representations of small parts of the input, whereas at upper layers generate more abstract descriptors of larger areas.

Shared Filters and Feature Maps.

In CNNs, a unique set of weights and bias (kernel or filter) is shared across the entire input field creating a feature map as can be seen in Figure 1(b). This means that all the neurons (or units) in a given convolutional layer will respond in the same way to the same feature over the previous layer, for example a vertical edge. This is done because it is highly likely that a learnt feature would be useful at other places of the image.

The main consequence of sharing these kernels, is that the feature can be detected regardless its position in the input field, obtaining the translation invariance property that convolutional neural networks have and that hand-crafted descriptors struggled to get. Still, the described structure would not be practical for doing robust image recognition, as it can only detect just a single kind of feature. More than one feature map is needed therefore, enabling to learn more kernels and obtaining a much more robust network, as it will be detailed in Section 3.2.2.

Another important consequence, is that CNNs are able to reduce dramatically the number of parameters learnt at each layer in comparison with a fully connected neural network. Following the previous example of a $28x28$ pixels input image and setting a local receptive field of $5x5$, the output of a convolutional hidden layer would have $24x24$ units. These units result from moving 23 times the reception field over the whole input image. Nevertheless, in this case the CNN only need to learn one filter (vector of weights and a bias) for generating a full feature map, so that $5x5 + 1 = 26$ parameters. Given the case that we choose to have for example 20 of those feature maps in the first CNN hidden level, we would need to learn $26x20 = 520$ parameters. In order to compare, a classical neural network first layer with 20 neurons would require for the same example $28x28x20 = 15680$ weights plus another 20 biases, thus 15700. This substantial reduction of weights of the CNN, leaves space to create bigger and deeper networks with several layers.

Pooling.

The last main difference of CNNs with respect to standard neural networks is the existence of pooling layers. The introduction of the pooling step aims to simplify the information at the output of the convolutional layers by reducing it.

As the pooling step can be seeing like a special layer having its own entity inside the structure of a CNN, we will provide a more detailed description of it in Section 3.2.2.

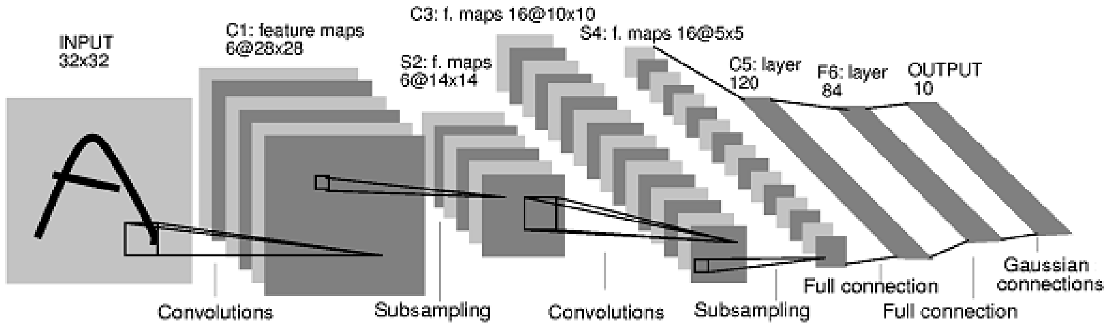


Figure 2: Depiction of a LeNet model widely used for classification problems. In the first step, a convolutional layer (C1) is applied to the input with a kernel size of $5x5$, producing 6 feature maps of size $28x28$. Next, a $2x2$ max-pooling operation (S2) is performed, reducing the size of the feature maps to $14x14$. This process (convolution+pooling) is repeated with the same kernel sizes, but in this occasion creating 16 feature maps in the convolution layer (C3). The final layers correspond to Fully connected ones and work in a similar way as classical neural networks do [13]. Finally, for classification problems, it is common to introduce a last layer obtaining the prediction for the known classes (for example, for classifying hand-written digits this layer would have 10 outputs).

3.2.2 Basic Convolutional Neural Network Structure

One of the earliest and famous Convolutional Neural Networks receives the name of *LeNet5* [18], after one of its authors, Yann LeCun. Since it firstly appeared, several other authors and CNNs have been based on it, creating great variety of modifications. However, its core structure of alternating convolutional layers with max-pooling ones has been maintained in most of the new approaches. A graphical illustration of a simpified LeNet model network is shown in Figure 2, which will serve as example for detailing CNNs main structure and characteristics later in this section.

Basically, LeNet consist on an input layer followed by 5 hidden layers and two fully connected layers at the end performing the final classification. The hidden layers are a sequence of a convolution layer (C1) followed by a max-pooling subsampling layer (S2). This structure is repeated one more time (C3 and S4). The last hidden layer (C5) corresponds to another convolution layer that reduce the 3 dimensions of the whole set of feature maps of S4 to a two-dimensional vector enabling to link to the fully connected layers (F6 and output soft-max layers). Detailed information of the main structural layers in a CNN is presented next.

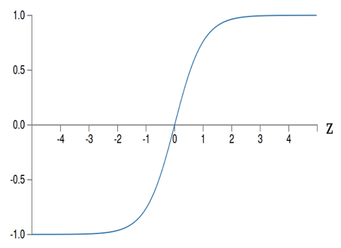
The Convolution Layer.

The output feature map of a convolution layer is calculated as:

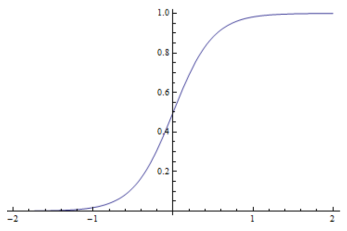
$$a_{x,y} = \sigma\left(b + \sum_{f_1}\sum_{f_2} w_{f_1,f_2} \cdot a_{j+f_1,k+f_2}\right)$$

where $a_{x,y}$ is the output unit; b is the shared value given to the filter bias; f_1 and f_2 goes from 0 respectively to the height and width of the filter used, and thus w_{f_1,f_2} are the the learnt values for the kernel weights; k, j is the top-left corner of the local receptive field in the input map; $a_{j+f_1,k+f_2}$ are the input values under the filter patch (input activation $a_{x,y}$); and finally σ is the neural activation function providing the non-linearity.

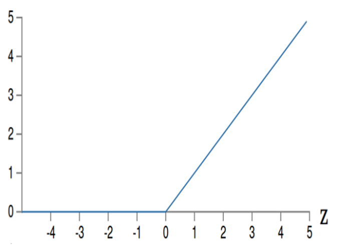
Kernel size ($f_1 x f_2$) values vary greatly, usually based on the dataset. For example, for the MNIST-sized images ($28x28$) best results are usually obtained with an 5x5 range on the first



(a) Hyperbolic Tangent



(b) Sigmoid Function



(c) Rectified Linear

Figure 3: Commonly used activation functions.

layer. Natural image datasets (often with hundreds of pixels in each dimension) tend to use larger first-layer filters of shape $12x12$ or $15x15$.

The purpose of the activation function is to introduce non-linearity into the network, so that non linear models can be learnt. Commonly used activation functions are the hyperbolic tangent (tanh) function, or the sigmoid one which can be seen in Figure 3(a) and 3(b) respectively. However, a new activation function called the *rectifier* has been recently adopted, represented in Figure 3(c). Units using this kind of non-linearity are commonly known as Rectified Linear Units (ReLU), and its outputs are calculated as $f(x) = max(0, w \cdot x + b)$, being w and b a defined weight and bias respectively.

The main advantages and drawbacks of using ReLU units have been studied in several articles such as [19]. Other works as [20] or [21], have found that the ReLUs improve considerable the performance of CNNs for image recognition. This is partly because ReLU units require only comparison, addition and multiplication operations, so that it allows for faster and effective training of deep neural architectures on large and complex datasets. Another important advantage is that ReLU units are able to provide a sparse activation on initializing for a network (as only about the 50% of the units would be activated with non-zero output).

The Pooling Layer

As stated in the previous section, one of the convolutional neural networks special concepts is pooling. Pooling layers were designed as a form of non-linear downsampling for progressively reducing the spatial size of the feature maps and hence, reducing the amount of parameters and computation in the network. Moreover, the pooling operation provides a form of translation invariance. Intuition says that once a feature has been found, its exact location is not as important as its rough location relative to other features, so in this way we are allowed to keep only the most activated unit.

There exist several pooling procedures, but the most common one is *max-pooling*. In max-pooling, each pooling unit simply outputs the maximum activation within a lxl input region, being l a integer value (typically 2). Other advanced pooling techniques exist. One example is L2 pooling, which instead of the maximum activation, takes the square root of the sum of the squares of the activations on the defined pooling region. Another recent example is Fractional Pooling [22], which is having quite good results by smoothing the rapid reduction in spacial size of max-pooling layers allowing to a special lxl region, where l can be a non-integer value.

The Upper Layers

Convolutional Neural Networks perform the high-level reasoning in the upper layers via fully connected layers. The units in a fully connected layer are connected to all the feature maps in the previous layer, as seen in regular Neural Networks between neurons. Therefore, their activations can be easily computed with a matrix multiplication followed by a bias offset.

The Upper layers also include the loss layer, normally the last one of the network. It specifies how the network training penalizes the deviation between the predicted values and the true ones. Depending on the task various loss functions may be used, being the most common Soft-max, which predicts a single class between a set of mutually exclusive ones. Other loss functions exist such as sigmoid cross-entropy loss, or euclidean loss. The former is commonly used for predicting n independent classes by probability values in the range of [0,1]. The latter is most used for performing logistic regression to real-valued labels in the [-inf,inf] interval.

3.2.3 Rise of Convolutional Neural Networks

In the past few years, CNNs have experimented a huge revival after being applied very successfully to solve extremely difficult computer vision problems. From year 2011 there have been numerous breakthroughs achieved by these networks, showing that an important transition is taken place in the research community. Several factors are responsible of this new success of CNNs.

Firstly, there have been new advances providing much faster developing and training tools for Convolutional Neural Networks. Powerful hardware units for GPU computation has come along with new software libraries (e.g. *CUDA* and *CuDNN*) enabling the easy exploitation of this hardware. In this way, several frameworks have also been created providing researchers with fast and easier ways of developing and training CNNs, as for example *Caffe*, *Torch*, or *MatConvNet*.

Secondly, the availability of new labelled databases providing extensive training sets with millions of samples allows to train models more exhaustively. Within this datasets, several competitions are carried out for different tasks like object classification and detection in which CNNs are surpassing by far standard computer vision approaches. A brief summary of some of the best known datasets and competitions that have contributed to the CNNs expansion are presented in Table 1.

Finally, the use of better regularization and normalization techniques are reducing the tendency of the networks to overfit. An example of the first ones is Dropout ([23], [24]), which consists on removing individual neurons at random during the training step. It reduces complex co-adaptations of neurons, forcing them to learn more robust features without depending on other units, as they suddenly may will not connected. In CNNs, Dropout was firstly used in [25] and is applied only in the fully connected layers because the convolutional layers already provide protection against overfiting by sharing the wheights, as commented in Section 3.2.1. An example of the second group, comes with batch normalization [26], which allows to use a higher learning rate and therefore to train faster (by normalizing the network parameters with each mini-batch) and therefore reduce the tendency to overfit.

3.3 Convolutional Neural Networks Models

Convolutional Neural Networks ability for building multiple layers of abstracted features seems to be fundamental to sense and understand our world, which is making CNNs a very hot research topic.

Many architectures are recently been proposed for different computer vision purposes such as feature extraction, object recognition and detection, or scene understanding. However, some of them are only slightly modifications or tweaked versions of other well known networks, making it difficult to track them or to keep an updated list.

Nevertheless, we present here a brief summary of the most famous CNNs architectures that have served as base for many other authors, along with its main characteristics and achievements:

Recognition and Classification Commonly used DataBases				
Name	About	Classes	Num. Images	Image Size
MNIST	Handwritten digits in gray	10	60K+10K	28x28
CIFAR-10	Objects in Color	10	50K+10K	32x32
CIFAR-100	Objects in Color	100	50K+10K	32x32
STL-10	Objects in Color	10	50K+80K	96x96
SVHN	Street View House Numbers	10	>73K+>26K	32x32
ILSVRC	High-res. Object images	1000	1.2M+150K	$\sim 482x415$

Object Detection Commonly used DataBases				
Name	About	Classes	Tr. Images	ROIs
PASCAL VOC	Objects in real scenes	20	11530	27450
Caltech Ped.	Pedestrians from driven vehicle	1	250K	350K/2300
TUD Ped.	Pedestrians + movement info.	1	1092	1776
KITTI	Real Driving Environment	3	7481 +7518	80.256

Table 1: Most used datasets for object recognition and detection. In the number of images field, the first value corresponds to the training set whereas the second to the test set. The STL-10 dataset, provides an extra 100K set of unlabelled images for unsupervised learning. The SVHN dataset provides also original images with bounding boxes (for detection purposes) and include >531K extra labelled, somewhat less difficult samples for extra training data. PASCAL Visual Object Recognition was a competition that run from 2005 to 2012 mainly for object detection and segmentation, after it, ILSVRC has also included object detection tests. The Caltech dataset differentiate between 2300 unique pedestrians. The TUD Pedestrians dataset include also information about movement and optical flow. Kitti dataset is a complete dataset with several options, but here we only refer to the object detection one, which focus on Car, Pedestrian and Cyclist classes.

- **LeNet**. Developed by Yann LeCun in the 1990s, it is the reference network in the field, as previously described in Section 3.2.2. It was the first successful application of Convolutional Networks to real computer vision problems, mainly to read zip codes and digits between others.

- **AlexNet**. Developed by Alex Krizhevsky *et al*, AlexNet [25] represented the real breakthrough of Convolutional Networks in Computer Vision. It was submitted to the ILSVRC challenge in 2012, winning it with a top-5 accuracy of 84.7% which was by far better than the 73.8% achieved by the second-best contest entry. AlexNet has a basic architecture similar to the LeNet one, but is deeper and bigger. It contains about $650K$ neurons, $60M$ parameters and $630M$ connections. Its main difference is that uses Convolutional Layers stacked on top of each other, which differs from previously common approaches that has a single convolution layer immediately followed by a pooling layer. It contains 7 hidden layers, the first 5 being convolutional (some with max-pooling) and the next 2 layers being fully connected. On top of it, has a 1000-unit soft-max output layer as the output classification for the 1000 image classes of the ILSVRC.

- **GoogLeNet**. Another noteworthy CNN architecture can be found at [27] under the name of *Inception.* A particular implementation of it, GoogLeNet, won the object detection task of the ILSVRC 2014, obtaining an incredible top-5 accuracy of 93.33%. Its main contribution is the development of the Inception Modules, inspired by the Network-in-Network work in [28]. This approach dramatically reduced the number of parameters in the network ($4M$, compared to AlexNet with $60M$), allowing to have deeper structures. GoogLeNet contains 22 parametrized layers, and combines the use of standard convolution+pooling layers with their inception modules. Moreover, concerned about the vanishing gradient problem [1], it add auxiliary classifiers connected to some intermediate layers to encourage

[1]A problem found when training neural networks with gradient-based methods and backpropagation. As traditional activation functions such as tanh or sigmoid have gradients in low ranges (-1, 1) or (0, 1), and backpropagation computes gradients by the chain rule (multiplying n of these small numbers to compute gradients of the "front" layers in an n-layer network), the gradient decreases exponentially with n and the front layers are trained very slowly.

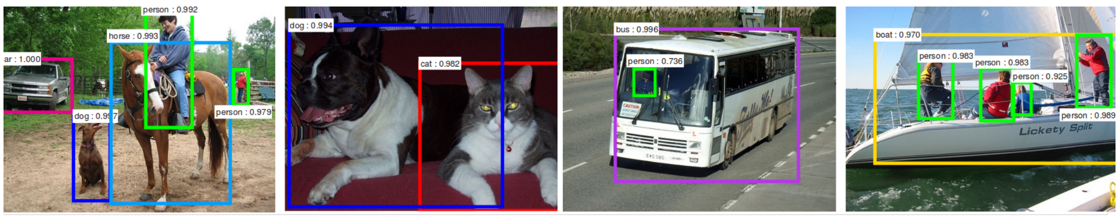


Figure 4: Examples of objects detected and classified with CNNs. Image extracted from the *Faster R-CNNs* [30] work of Shaoqing *et al.*

discrimination at lower stages, increase the gradient propagated and provide additional regularization.

- **VGGNet**. The VGGNet [29] created by Simonyan and Zisserman presents a very deep although very simple and homogeneous structure. It obtained the second-best position in ILSVRC 2014 with a total of 16 convolution layers plus fully connected layers at the end. Simplicity is provided by applying from the beginning to the end blocks of 2 convolutions and one max-pooling layer. Their main contribution was to show that the depth is a critical component for good performance in a CNN. Despite its lower classification performance (in comparison with GoogLeNet), it has been found that the VGGNet features outperform those of their rivals in multiple transfer learning tasks. This generalization capacity has made of VGGNet to be one of the most preferred choices in the community when extracting CNN features from images. However, a drawback of VGGNet is that it is more expensive to evaluate and uses a lot more memory and parameters, as it has around 140M parameters).

As deep architectures, CNNs have the need of lot of memory which sometimes is a big inconvenience working with GPUs. A rough estimation of the size of a network can be done knowing the number of parameters used, multiplying those by 4 (or 8 in double precision) and then dividing consecutively by 1024 to get the size in KB, MB or GB. In this way for double precision, an AlexNet model would require about 450MB of memory, GoogLeNet only around 30MB and VGGNet around 1GB.

3.3.1 Convolutional Neural Networks for Object Recognition and Detection

One of the most direct uses of CNNs is to classify and detect objects in images. This last problem, is a more challenging one (as shown in Figure 4), and requires to detect in the same image several objects instances that can vary on size and pose, or even be occluded. Some of the datasets, challenges and competitions summarized in Table 1 (as well as released pretrained models of the previously commented CNNs) have contributed to the development of algorithms and other CNNs architectures able to solve the object detection problem.

Initial approaches use well known multiscale and sliding window methods in combination with CNNs-extracted features and a final classifier. However, Sermanet *et al* showed in their Overfeat work ([31]) that *end-to-end* trained CNN architectures can also we designed to produce an integrated approach to object detection, recognition and localization. They explored CNN special characteristics of location invariance and weights sharing to develop an inherently efficient sliding window approach. They also introduce a novel method that learn to predict object boundaries to create bounding boxes, all in the same CNN architecture.

On the other hand, the recent advancement of region proposal methods (e.g, [32]) have established new successful CNNs architectures. An important one, was proposed by Girshick *et* *al* in [33], as what is called region-based CNNs (R-CNNs). It starts with a pre-processing region proposal step that outputs 2000 proposals. Next, it uses a pre-trained AlexNet classification network to extract a 4096 feature vector for each of the regions. Finally, it classifies each region with a category-specific linear SVMs and with the results they fine tune the CNN end-to-end for detection.

Further works on Object Detection with CNNs has focus mainly on reducing the expensive computations of R-CNNs, which has been achieved succesfully by sharing the convolutions across proposals [34], [35], [30]. The SPPnet work of He *et al* [35] speeds up R-CNNs by moving the proposals warping and its interpretation after the last convolutional layer. On the contrary, work done by Shaoqing *et al* in [30], directly propose a fully convolutional network able to generate by itself the region proposals by adding two additional convolutional layers to the network.

3.3.2 Convolutional Neural Networks for Feature Extraction

As stated in the previous sections CNNs are able to obtain robust image features, from low level layers detecting for example edges, to more sophisticated structures and abstract concepts at higher layers. This, in combination with its translation invariance property makes CNNs good candidates to act as simple substitute for standard hand-crafted descriptors described in Section 3.1. Recent work of Yosinski *et al* [36] helps on this topic with a studio about feature transferability in CNNs, this is, using features obtained from a pre-trained network as descriptors for other utilities. Moreover, Zeiler *et al* [37] address the problem of interpreting what the kernels of a CNNs learn at the different layers.

The idea of using CNNs as direct robust feature generators has been harnessed by several authors for object detection or patch matching by including convolutional layers in their algorithms pipeline, such for example the works of [33] or [38] respectively. Besides, other authors like Simo-Serra *et al* [39] use CNNs (specifically siamese networks), to create a 128-D descriptor that can be used directly as a drop-in substitute for SIFT. Its descriptor is demonstrated to be efficient and generalize well against scaling and rotation, perspective transformation, non-rigid deformation, and illumination changes.

Other recent CNNs-based approaches, are more interested in per-pixel prediction problems, being applied for example to semantic segmentation or Optical Flow. From these, useful low level information can be obtained to be used at further scene understanding.

Semantic Segmentation with CNNs

Semantic segmentation has been a long-time known problem in Computer Vision, aiming to assign each of the pixels of an image to a set of pre-defined object (semantic) categories as can be seen in Figure 5. Segmentation algorithms typically have three main components; one modelling the local appearance of objects (clustering image pixels into homogeneous regions), other taking into account and enforcing the local consistency of the labels between locations (grouping the regions into semantically meaningful parts), and the last one enforcing global consistency at region or image level (formulating the problem as a minimization problem over the space of labelings/segmentations). These separated components are commonly integrated into unified probabilistic methods, e.g. Conditional Random Fields (CRF) [40], obtaining good performances. But these integrated models comes with a high computational cost.

Semantic segmentation problem has been recently tackled with CNNs. In this way, several works exist, as for example [41], [42], [43]. Remarkably is the work of Long *et al* in [43], which proposed Fully Convolutional Networks (FCN) as special CNNs able to take input images of any size and produce correspondingly-sized outputs with efficient inference and learning. In this case, they showed that FCNs trained end-to-end, pixel-to-pixel on semantic segmentation

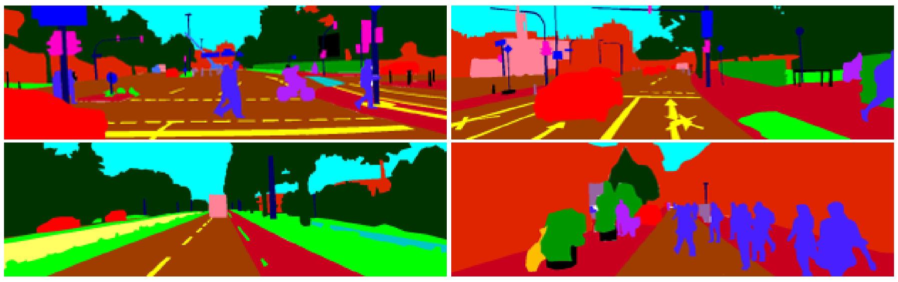


Figure 5: Ground truth examplesof the Kitti Dataset for semantic segmentation.

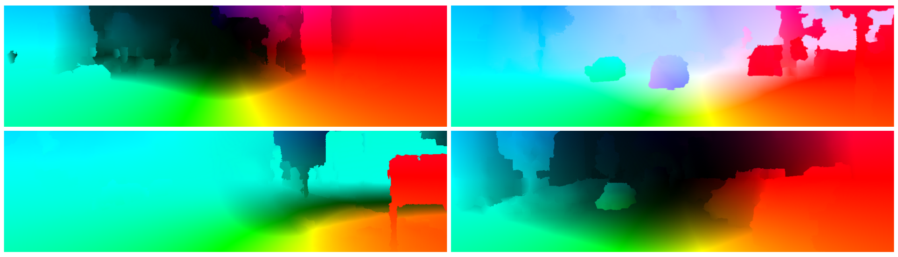


Figure 6: Ground truth examples of the Kitti Dataset for optical flow estimation.

exceeded without further requirements the state of the art algorithms.

Another noteworthy work is [44], which aims to solve the poor pixel-feature location at the high layers of a CNN to produce semantically accurate predictions and detailed segmentation maps. They manage to do this, while being computationally efficient, by combining responses from the CNNs with a fully connected Conditional Random Field in their *DeepLab* system. Very recent approaches like [45], try to produce delineated outputs on semantic segmentation, by introducing a new end-to-end form of CNN able to combine internally CNNs and CRF. They called this network CRF-RNN, and can be plugged as a part of a CNN to obtain a new deep network with the desired properties of both CNNs and CRFs.

Optical Flow Estimation with CNNs

As a method to estimate the motion on a scene, optical flow requires to find correspondences between two input images, not only by means of per-pixel localization, but also by matching image feature representations. Figure 6 shows some Optical Flow examples extracted from the Kitti dataset. Many challenges arise for optical flow problem in realistic dynamic environments, such as outliers (occlusions, motion discontinuities), illumination changes, and large displacements. However, those could also be addressed with CNNs as seen before.

Optical flow estimation, has been dominated by variational approaches since the work of Horn and Schunck [46]. Further research has focused mainly in alleviating the drawbacks of the method, introducing different improvements, [47], [48], [49], or managing large displacements [50].

Deep learning techniques were introduced in optical flow estimation by Weinzaepfel *et al* in their DeepFlow [51]. They realised that current descriptor matching approaches are based on rigid descriptors, creating implicit rigid motion hypothesis that do not fit for fast neither large motions. As a solution, they propose to firstly extract descriptors in non-rigid local frames by means of sparse convolutions and max-pooling, for later on performing dense matching in all image regions. They showed that their approach resulted to a significant performance boost on MPI-Sintel [52] and KITTI [53] optical flow datasets.

Per contra, there has been little work on fully estimating optical flow by using Convolutional Neural Networks. A good approach worth to mention is the Zbontar and LeCun [54] work, which train a CNN with a Siamese architecture that is able to predict similarity between image patches.

However, a recent method termed FlowNet and presented by Fischer *et al* in [55] creates an end-to-end CNN architecture able to learn to estimate horizontal and vertical flow fields directly from the image pairs. The authors proposed two different networks for solving the problem without requiring any handcrafted method for aggregation, matching or interpolation. The initial one is a simple end-to-end CNN architecture based on convolutional layers with an input of six channels from the two images. They also proposed a second approach with two different convolutional networks image inputs merged by a new *correlation* layer. This layer aims to help the network in the pixel matching process by performing multiplicative patch comparisons between the two feature maps obtained. The correlation operation is identical to a convolution operation, but instead of convolving data with a learnt filter, it convolves data with other data (so no weights are needed). As a final step, both networks perform a refinement process by means of upconvolutional [56] [2] layers merged with the coarse predictions. In addition, they perform an optional variational approach in order to do a final refinement of the network predictions, obtaining results comparable to state of the art algorithms.

3.4 Scene Understanding

Scene understanding aims to provide computers with human capabilities for accurately and comprehensively recognize and understand our complex visual world. It is a core problem of high level computer vision aiming to solve questions such as What is happening?, which elements are participating in the action?, what should I do next?. To solve these questions a global understanding of the environment is needed. As humans we are able to understand scenes at a whole by localizing and recognizing different objects (object classification and detection), delineating their limits (semantic segmentation), and/or knowing the dynamics and movements (optical-flow). In such manner, scene understanding can be seen as a wrapping of all the previously commented low level features extracted, to perform a complete interpretation of the scene.

As we have seen in previous sections, these computer vision problems have been commonly explored independently. However, they could benefit from a jointly approach, as all of them are very related at high level. For example, if we know the objects under studio, segmentation could be done easier with the prior awareness of the shape; we could better detect and localize objects if we know the semantic regions - boats on water, cars on roads; movement estimation would be more accurate if the object is know; etc.

Several works have been done in this line with holistic approaches, demonstrating that combining some of the previous independent solutions can outperform individual results [57], [58], [59], [60], [61]. These holistic approaches are commonly based in building probabilistic models such as CRFs, for integration and reasoning at pixels, regions or object levels (including their attributes).

Other approaches try to keep the problems separated, as for example the work of Cadena *et al* [62]. They propose a distributed scene understanding system that treat it perception task as software modules. These, can be activated or deactivated without impairing the rest of the system, allowing the communication between modules to improve their respective performances.

Following the same progression as other computer vision problems, scene understanding is started these days to be analysed with Deep Learning technologies. In this way, a very recent

[2] Upconvolutional, or deconvolutional layers, have been used in different works (as [37]) and performs the opposite operation than the convolution

work of Shao *et al* [63] propose a convolutional neural network model to jointly learn appearance and motion features and effectively combine them.

4 Work Plan

In this chapter we will present the different tasks to be carried out for the accomplishment of the objectives of this research plan, as well as an intended Gantt chart for their execution.

- **Task 1 - Literature Review.**
  This task aims to obtain a general overview on the state-of-art about Convolutional Neural Networks by reviewing its main concepts, structures and characteristics. We will also analyse its successful performance beating standard feature-based approaches in common computer vision problems such as object classification and detection as well as extracting low level features like optical flow and semantic per-pixel segmentation.

- **Task 2 - Low level information extraction using CNNs.**
  The aim of this task is to exploit CNNs inherent properties in order to obtain low level features from the scene. We will focus on developing CNNs for estimating the optical flow between images as well as for performing per-pixel semantic segmentation.

  Task 2.1 - Explore and develop CNN architectures for computing optical flow.

  Task 2.2 - Explore and develop CNN architectures for performing per-pixel semantic segmentation.

  Task 2.3 - Train proposed networks only with synthetic data and explore its generalization capacities as well as the transference of learning when retrained with real data.

- **Task 3 - Scene Understanding based on CNNs.**
  Within this task we aim to design and develop new CNN architectures able to perform multimodal feature integration. We aim to exploit the low level features obtained previously and combine them in order to solve higher level computer vision problems such as scene understanding.

  Task 3.1 - Propose and develop end-to-end trainable CNN architectures able to integrate the different extracted features with appearance RGB images.

  Task 3.2 - Investigate on adding depth features (dense depth from cameras or sparse data provided by laser scanners) into the proposed CNN architectures. How to use unknown sized inputs into CNNs.

  Task 3.3 - Explore techniques for introducing specific domain and application knowledge into the networks to make the approaches efficient for autonomous driving.

- **Task 4 - Research Stay:**
  Research stay in an international laboratory.

- **Task 5 - Final Thesis Document:**
  Writing, deposit and defense of the final doctoral thesis.

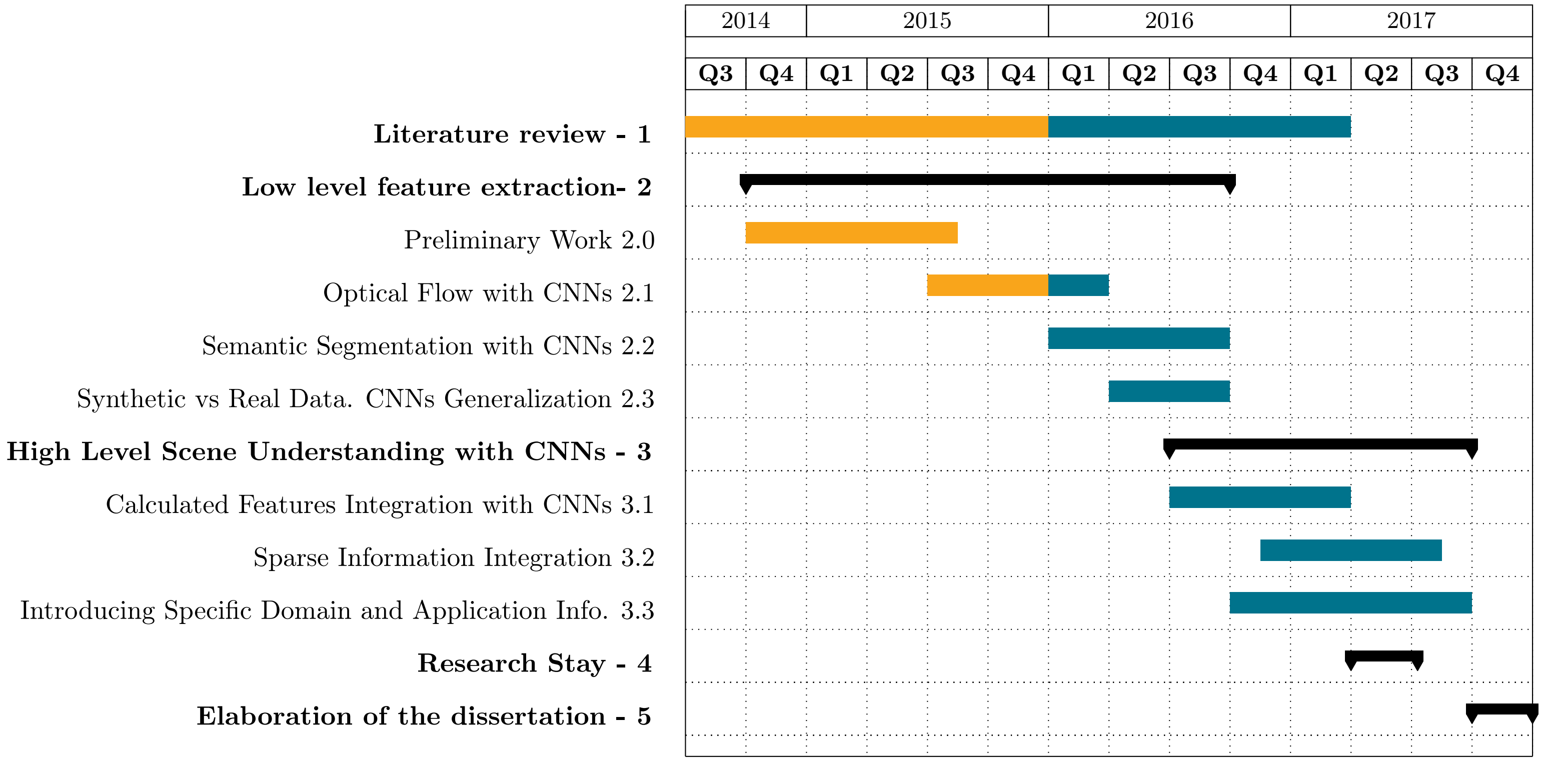


Figure 7: Work plan of the proposed work

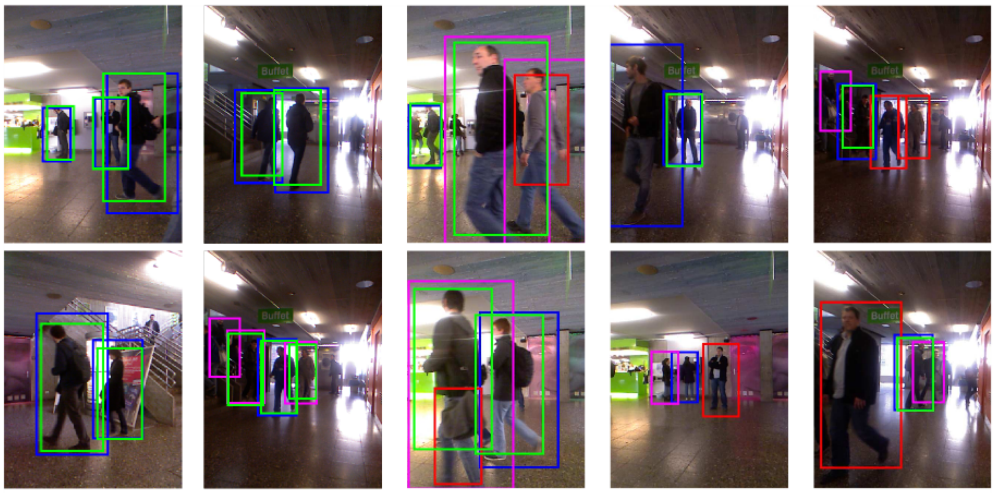


Figure 8: Sample images of our Boosting-based feature integration approach for people detection in RGB-D image spaces. Full people annotations are indicated by blue boxes. Magenta rectangles correspond to occluded people. Correctly detections (true positives) are represented in green. False positives are shown by red boxes. Columns 4 and 5 shows some *false* false positives (correct detections but not annotated in the dataset), which really penalises our detector. However we achieve very good results around a 89% of EER, beating other methods.

5 Preliminary Results

Preliminary work on this thesis has focused on the studio of the state of the art, and investigations tackling traditional ways of facing some of the problems we have set as objectives in this research plan.

On one hand we have explored traditional feature integration methods for building a strong classifier, so that obtaining a better and practical notion of how they work for further comparison with CNNs based methods. For doing this, we proposed in [64] a new and fast method for detecting people in dense appearance and depth images by building a strong classifier able to learn to integrate the most discriminative features from both spaces. We use as features Random Ferns and by means of an AdaBoost we build a strong classifier able to integrate the most discriminative features from both spaces in a final robust model. We validate our approach by tackling the problem of people detection in a challenging RGB-D database outperforming state-of-the-art approaches (see Figure 8) in spite of the difficult environment conditions, obtaining around an 89% of Equal Error Rate.

On the other hand we have explored the problem of detecting moving objects with the sparse information provided by single-layer range scanner, as these sensors are typically used in autonomous vehicles technology. We wanted to know the possibilities and difficulties of standard approaches for detecting objects by building features manually within this sparse data. The conclusion of this work is that single-layer range-laserscanner by its own provides not enough information for performing high level semantic scene understanding, and therefore additional sources of information such as appearance RGB images are needed.

This preliminary research is presented in [65], where we approached the moving object detection problem with a standard detection+tracking solution, focusing on the detection part. In order to deal with the very little information obtained from the used laser-scanner, we realised that only geometric features such as lines and corners could be extracted from the objects (clusters of points) detected. We have also realised that one of the main problems when working with one single laserscanner is the partial observability of objects and therefore, the absence of any reference point that could be tracked over time. In this work we propose a reference propagation algorithm that aims to solve this problem by propagating singular object geometries found. However, we consider that this problem could be better solved with the introduction of additional appearance information such as RGB images. Moreover, due to the robust results and high performance that CNNs are obtaining these days, we will in this thesis research on how to integrate both laser features and appearance as well as detect moving obstacles by making use of these networks.

5.1 Preliminary Publications

The preliminary work has led to the following already published congress results:

- **[64]** V. Vaquero, M. Villamizar, A. Sanfeliu. *Real time people detection combining appearance and depth image spaces using boosted random ferns.* In Robot 2015: Second Iberian Robotics Conference, Vol. 418 of Advances in Intelligent Systems and Computing, 2015, pp. 587–598.
- **[65]** V. Vaquero, E. Repiso, A. Sanfeliu, J. Vissers, M. Kwakkernaat. *Low cost, robust and real time system for detecting and tracking moving objects to automate cargo handling in port terminals.* In Robot 2015: Second Iberian Robotics Conference, Vol. 418 of Advances in Intelligent Systems and Computing, 2015, pp. 491–502.

References

[1] D. J. Fagnant and K. Kockelman, “Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations,” *Transportation Research Part A: Policy and Practice*, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *arXiv preprint arXiv:1502.01852*, 2015.

[3] A. Vedaldi and K. Lenc, “Matconvnet – convolutional neural networks for matlab,” in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015. [Online]. Available: http://www.vlfeat.org/matconvnet

[4] D. Lowe, “Object recognition from local scale-invariant features,” in *International Conference on Computer Vision (ICCV)*, 1999.

[5] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Lecture Notes in Computer Science*, 2006.

[6] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, 2004.

[7] H. Zhang and Q. Hu, “Fast image matching based-on improved SURF algorithm,” in *International Conference on Electronics, Communications and Control*, 2011.

[8] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition (CVPR)*, 2001.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[10] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," in *Intelligent Vehicles Symposium (IVS*, 2006.

[11] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *International Conference on Robotics and Automation (ICRA)*, 2014.

[12] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[13] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall PTR, 1998.

[14] Z.-R. Wang, Y.-L. Jia, H. Huang, and S.-M. Tang, "Pedestrian Detection Using Boosted HOG Features," in *IEEE International Conference on Intelligent Transportation Systems*, 2008.

[15] I. Laptev, "Improving object detection with boosted histograms. ," *Image and Vision Computing*, 2009.

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.

[17] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, 1989.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, 1998.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning (ICML)*, 2010.

[20] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *International Conference on Computer Vision (ICCV)*, 2009.

[21] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[22] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

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

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, 2014.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014.

[28] M. Lin, Q. Chen, and S. Yan, "Network In Network," *arXiv preprint arXiv:1312.4400*, 2013.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," in *International Conference on Learning Representations (ICLR)*, 2013.

[32] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, 2013.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[34] R. Girshick, "Fast r-cnn," in *International Conference on Computer Vision (ICCV)*, 2015.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision (ECCV)*, 2014.

[36] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[37] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision (ECCV)*, 2014.

[38] A. Penate, L. Porzi, and F. Moreno-Noguer, "Matchability prediction for full-search template matching algorithms," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2015.

[39] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *International Conference on Computer Vision (ICCV)*, 2015.

[40] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning (ICML)*, 2001.

[41] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.

[42] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *European Conference on Computer Vision (ECCV)*, 2014.

[43] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[44] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *International Conference on Learning Representations (ICLR)*, 2015.

[45] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional random fields as recurrent neural networks," in *International Conference on Computer Vision (ICCV)*, 2015.

[46] B. K. Horn and B. G. Schunck, "Determining optical flow," in *Technical symposium east*. International Society for Optics and Photonics, 1981.

[47] A. Bruhn and J. Weickert, "Towards ultimate motion estimation: combining highest accuracy with real-time performance," in *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[48] A. Wedel, D. Cremers, T. Pock, and H. Bischof, "Structure - and motion - adaptive regularization for high accuracy optic flow," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[49] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[50] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2011.

[51] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large Displacement Optical Flow with Deep Matching," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[52] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision (ECCV)*, 2012.

[53] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, 2013.

[54] J. Žbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," *arXiv preprint arXiv:1409.4326*, 2014.

[55] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *International Conference on Computer Vision (ICCV)*, 2015.

[56] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *International Conference on Computer Vision (ICCV)*, 2011.

[57] L.-J. Li, R. Socher, and L. Fei-Fei, “Towards total scene understanding: Classification, annotation and segmentation in an automatic framework,” in *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[58] S. Gould, T. Gao, and D. Koller, “Region-based segmentation and object detection,” in *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[59] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr, “What, where and how many? combining object detectors and crfs,” in *European Conference on Computer Vision (ECCV)*, 2010.

[60] A. Geiger, C. Wojek, and R. Urtasun, “Joint 3d estimation of objects and scene layout,” in *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[61] J. Dong, Q. Chen, S. Yan, and A. Yuille, “Towards unified object detection and semantic segmentation,” in *European Conference on Computer Vision (ECCV)*, 2014.

[62] C. Cadena, A. Dick, and I. D. Reid, “A fast, modular scene understanding system using context-aware object detection,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

[63] J. Shao, K. Kang, C. C. Loy, and X. Wang, “Deeply learned attributes for crowded scene understanding,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[64] V. Vaquero, E. Repiso, A. Sanfeliu, J. Vissers, and M. Kwakkernaat, “Low cost, robust and real time system for detecting and tracking moving objects to automate cargo handling in port terminals,” in *Robot 2015: Second Iberian Robotics Conference*, ser. Advances in Intelligent Systems and Computing, 2015.

[65] V. Vaquero, M. Villamizar, and A. Sanfeliu, “Real time people detection combining appearance and depth image spaces using boosted random ferns,” in *Robot 2015: Second Iberian Robotics Conference*, ser. Advances in Intelligent Systems and Computing, 2015.