* 1. **– Introduction**

Humans are unmistakably the most important components of a machine’s environment. Whenever people are involved in a process with any associated risks, there is a great number of special security rules that must be followed in order to assure their safety. Visual detection of humans is a field with an extensive range of applications such as robotics, entertainment, surveillance, care for the elderly and disabled, road safety and others. In all of these, the knowledge of the presence of a person allows the equipment with whom it’s interacting to act accordingly, be it sounding an alarm, stopping an operation, or any other action. The benefits of this become obvious when we think about a car being driven in an urban scenario. If the circumstances are such that the driver is always aware of the surrounding pedestrians, danger of accidents involving them would most likely decrease dramatically. One could even think of a safety mechanism that refrains a driver's action in a dangerous situation for a pedestrian.  
In the European Union, 21% [reference] of all traffic fatalities are pedestrians, indicating that we’re looking at a matter of great importance. Motivated by this, many researchers have devoted much work in developing algorithms for visual human detection, leading to extraordinary improvements in the past decade. Despite those significant improvements, there is still much room for progress due to the challenging nature of the problem. Issues like varying lighting conditions or uncertain pedestrian postures require robust solutions in order to overcome those difficulties.  
In this work, an algorithm capable of visually detecting pedestrians achieving state-of-the-art detection rate is presented. The objective was to build a base detector to be inserted in the ATLAS project [reference], of the Department of Mechanical Engineering of the University of Aveiro. This is an ongoing team project that started with the aim of participating in autonomous mobile robots competitions and has since then is grown into real road vehicles (Figure 1.1) with the goal of developing new Advanced Driver Assistance Systems (ADAS).

Figure 1.1 – ATLAS car

Visual detection of pedestrians is an obvious and paramount feature that had yet to be incorporated in the project, making this work a significant contribution. The work presented in this document constitutes a base application that has still much room for improvement in the future, as this is a subject in constant development.

* 1. **– Problems**

Visual pedestrian detection is a challenging task with a set of complex problems to overcome. In this section, an overview of some common problems associated with detecting pedestrians in individual monocular images will be presented.

Computer Vision (CV) is a technology that has grown in presence on many fields of society over the past two decades. In industry, product inspection systems have significantly improved with the aid of CV by allowing inspection of parts at a major scale, a fact that lead to considerable advancements in the process of finding defects. In such environments, a careful setup is planned in order to facilitate the processing of the images outputted by the camera, since controlling the lighting level, background color and other external parameters is of utmost importance for an easy object segmentation, meaning, separating the object of interest from the background.

On the contrary, it is virtually impossible to control the external factors of the images where pedestrians must be detected, precluding the possibility of segmentation and causing the need to process cluttered, random images with huge amounts of information. The unpredictability of the location where a pedestrian might come into sight also mandates the analysis of the whole scene. Moreover, the varying nature of the lighting conditions caused either by changes in the daylight, or different weather conditions further hampers the task. Another typical problem that leads to relatively high miss rates is that pedestrians often appear partially occluded by other objects in the scene, such as trees, traffic signs, bikes, and even by other pedestrians. The uncertainty of their posture also constitutes a problem, since it is obvious that an up-right pedestrian has different properties in an image than one sitting down or leaning into another object.

In sum, the mission is to detect pedestrians that might or not be partially occluded, in unpredictable locations, assuming different stances, on cluttered scenes with varying lighting conditions. Such conditions demand highly robust algorithms which are typically heavy and unable to run at the frame-rates that this task demands. Figure 1.2 attempts to illustrate some of this problems.



Figure 1.2 – Varying lighting conditions, partial occlusion and different postures are some of the problems associated with pedestrian detection. Images taken from the INRIA dataset (Dalar and Triggs, 2005)

**1.2– State of the Art**

A great development has been made on the subject of visual pedestrian detection in the past two decades. On this section a compact description of some notorious contributions for this area will follow. This review will focus firstly on detectors with a sliding window approach, often seen as the most promising for low and medium resolution approaches. Secondly, some multi-sensor applications for ADAS will also be analyzed.

**1.2.1 – Sliding Window Detectors**

One of the first sliding window visual object detector attempted to describe an object class in terms of an over-complete dictionary of local, oriented multi-scale intensity differences between adjacent regions, also known as Haar Wavelets, and apply them to an example-based machine learning approach, where a model of an object class is derived implicitly from a training set of negative and positive examples (Papageorgiou et al., 2000). The specific learning engine that was used was a Support Vector Machine (Cortes and Vapnik, 1995) classifier, and results for car, faces and people detection tasks were shown [figures of results?]. Before this work, visual human detection had not yet been successfully tackled, as they [references?] would typically assume a number of restrictive assumptions in order to produce results.

Building upon Papageorgiou's ideas, (Viola and Jones, 2001) (VJ) proposed a method that extracted Haar-like features with an highly optimized approach due to the use of integral images, which is an image transformation that allows for rectangular sums of pixels to be computed by fast arithmetic operations. In addition to this, a learning mechanism based on the AdaBoost algorithm (Freund and Schapire, 1999) was utilized in order to select the most relevant features to perform classification, and a decision structure in the form of a cascade was built for efficient decision-making. This cascade works by evaluating sets of features that grow in complexity as a sample advances in the structure, an idea that stands upon the notion that a positive instance in an image is an extremely rare event. By rejecting most negative samples in the earliest stages of the cascade, this method, applied to face detection, was able to run at 15 frames per second (FPS) with a high success detection rate. This is a popular and widely spread approach that still serves as foundation for many modern detectors, and full implementations of the method were made available in software development tools such as OpenCV and MatLab.

Up until this moment, detection algorithms worked mostly on intensity images, a principle that changed when gradient-based features were introduced in the scope of pedestrian detection. Becoming widely known as Histogram of Oriented Gradient (HOG) (Dalal and Triggs, 2005), this method attempted to define a scene by dividing it into small spacial regions (cells), and accumulating for each one a local histogram of normalized gradient directions. These cells are combined over slightly larger and overlapping spacial regions (blocks), and each block is also locally normalized for better invariance to lighting conditions. The above-mentioned descriptors are then applied to a trained SVM based window classifier that identifies if a pedestrian is present in the scene or not. This contribution resulted in large gains when compared to intensity based methods and, since its introduction, the number of variants of HOG has increased to the point that nearly all modern detectors use some form of these features.

Interpretation of shape is also an important cue to the subject. In general terms, shape-based methods work by generating templates of the desired object and finding matches for them in visual data. The work developed by (Gravilla and Philomin, 1999) was one of the first to adopt this approach in the domain of pedestrian detection. It uses the Hausdorff distance transform and a template hierarchy to rapidly match image edges to a set of shape templates, and tests were made for pedestrian and traffic sign detection with satisfactory results. Still on this note, (Sabzmeydani and Mori, 2007) used gradient-based HOG-like features combined with an AdaBoost engine to learn head, torso, legs and full body shapes. In this approach two kinds of features are used for classification: the low-level features, which are simple and reminiscent to Haar-like features, and mid-level features that are learned part models for template matching. This method is documented to outperform HOG by a considerable margin.

Some researchers have used motion features to further improve detection results. The basic idea is that in an usual situation people are in motion, rather then sitting still. Therefore it is natural to think that if the circumstances are such that detection of motion is achievable, important clues as to the possibility of the presence of pedestrians will be found. It is, however, a challenging task to incorporate motion features into detectors given a moving camera. Given a static camera, (Viola et al., 2005) proposed a similar approach to their previous work, but applied to the result of the difference of two sequential frames, resulting in large performance gains. For non-static imaging setups, camera motion has to be factored out, as did (Dalal et al., 2006) when they attempted to model motion statistics based on an optical flow's (Fleet and Weiss, 2006) internal differences, thereby compensating for uniform motion locally.

Although HOG has not been outperformed by any single feature, some researchers hypothesized that assembling multiple types of features could provide important complementary information. To prove this, (Wojek and Schiele, 2008) combined Haar-like features, shapelets, shape context, and HOG features to compare the resultant detector with each of the features performing on their own, demonstrating that the combo outperforms any single feature detector. This framework was later extended to include the above-mentioned motion features in (Walk et al, 2010), further improving the detection results.

Using a different course of action, (Dollár et al., 2009) extracted Haar-like features from various channels, including gradient magnitude, LUV color channels and gradient magnitude quantized by orientation, providing a simple framework for integrating multiple feature types. In the author's approach, a large pool of features is extracted from random regions of the channels to guarantee a good characterization of the scene. A decision structure similar to the VJ's method is utilized with the purpose of selecting the most relevant features and performing efficient classification. This method became known as *Integral Channel Features*. A significant optimization was made to this algorithm when the authors hypothesized that features could be approximated at nearby scales with little sacrifice to results (Dollár et al., 2010). By eliminating the need to extract features at every scale, this algorithm is documented to perform multi-scale detection at 6 FPS and ranks among the best found in literature. A better still version of this framework was introduced in (Dollár et al., 2012), in which an even more efficient decision structure was proposed. In this brand new *Crosstalk Cascades* method, it is established that nearby decision windows have correlated responses. By creating a mean of communication between detector's responses, this method achieves similar detection rates as the *Integral Channel Features* while increasing speed by an order of magnitude.

Some authors have made an effort to modify learning engines to improve their speed, as did (Tuzel et al., 2008) by utilizing covariance matrices computed locally over a large diversity of features as object descriptors. The learning data doesn't lie on the vector space, thus improving the learning/testing performance. Non-linear SVM applied to sliding window detection was made possible when (Maji et al., 2008) found that the use of the learning algorithm with an approximation to the histogram intersection kernel lead to substantial gains in terms of speed.

**1.2.2– Multi-Sensor Detectors**

Visual data has great potential due to the richness of information it holds. However, it may be a challenging task to build a reliable detector to be implemented in an ADAS relying solely on camera output as a result of the problems discussed in the previous section. To overcome those difficulties, some try to ally different type of sensor information to create robust and more reliable detectors.

Infra-red image and laser data were used by (Fardi et al. 2005) to generate regions of interest where pedestrians might be at sight. A first-step classification is obtained by evaluating a set of descriptors based on the Euclidean distance of Fourier between objects and reference sets on infra-red visual data, a classification that is later refined by motion features acquired using egomotion sensors and optical flow.

On a different approach, radar, color and infra-red information is fused in (Marchal et al., 2005). In this work, hypothesis are generated through the evaluation of vision-based local histograms on edges, computed both on color and infra-red visual data. Neural Networks is the learning engine used for a preliminary classification, and its output undergoes further verification by a fusion between radar and tracking information.

Combining infra-red and visual spectra from the two camera types was proposed by (Bertozzi et al., 2006-2007). In the author's work, foreground segmentation is carried out by overlapping 2D and 3D information from both sensors and, finally, symmetry and template matching are used to classify, verify and refine final detections.

Laserscanner-based tracking of points was the strategy chosen by (Premebida et al. 2007) to generate candidate regions of interest for further analysis. Objects were defined by laser and visual features, and AdaBoost was utilized for generating responses.

These systems were built in scientific research environments where information is usually made accessible for anyone. That is not the case in industrial environments where, for commercial reasons, conducted research is kept in absolute secret, a circumstance that makes it hard to find reliable sources about the state of this technology in industry. It seems granted, however, that the first pedestrian detection system to be commercialized will be launched in 2014 by Mercedes and will be based on stereo camera images [reference].

**1.3 – Solution**

Although great advantages arise from fusing different type of sensor information, a multi-sensor approach also has its issues, such as difficulties in fusing and correlating different sensor data, more complex system implementations, accumulation of errors generated from different sensors, and others. Despite these problems, it is obvious that any real and full-functional pedestrian detector will, in all likelihood, require the use of multiple sensors as a result of the extremely demanding nature of the task in hands.

However, creating such complex application is a large engineering effort that requires a great deal of know-how, expensive equipment and time, especially when the application is being built from scratch. Although much useful equipment already exists in the laboratory, as well as a staff with a comprehensive knowledge and set of skills, building a full-functional, multi-sensory system was never the objective due to the short amount of time available to complete this work. The goal was to build a reliable, generic and simple vision-based pedestrian detection framework that leaves the possibility for future development and integration in more complex systems. It was decided then that the implementation of a sliding window algorithm was the way to go, since one can be modified to work with other sensors in future development.

In order to define a course of action, two premises were established: the implemented algorithm should rank among the best in terms of detection performance and should also leave space for future improvements. In respect of this, (Dollár et al., 2012) made an extensive survey of existing sliding window detectors, where 16 algorithms were compared against each other in a carefully designed evaluation platform. Out of all the evaluated algorithms, *Integral Channel Features (ChnFtr)* proved to be the most interesting for several reasons. Firstly, the only method that slightly outperforms it uses motion and gradient-based features, a computationally heavy approach that is documented to run ~50 times slower than *ChnFtr*, rendering it uninteresting. On the contrary, the authors of *ChnFtr* have largely improved its performance in later work, to the point of enabling multi-scale detection at 30 FPS. Secondly, the method provides a relatively simple framework in terms of code implementation when compared to other approaches. Since this implementation fitted perfectly with the proposed goal, it was the chosen way to go for this project.

A detailed explanation of *ChnFtr* is presented in chapter 3.

**2 – Development Tools and Experimental Setup**

To facilitate and standardize the development process of this project, three main tools were utilized. In this chapter, a description of those tools will be carried out, followed by how they integrate the experimental setup.

**2.1 – Robot Operating System**

The Robot Operating System (ROS) (Quigley et al., 2009) provides a software development framework that is designed for the creation of robot software. This application has several built-in components prepared to handle the output of different types of sensors, such as cameras, lasers, actuators, contacts and other common elements in a robotic environment.

ROS also allows for an easy to establish communication between different software modules (*nodes*), which permits the elaboration of an infrastructure that can communicate with any running processes. This communication works in three steps: first, a *node* advertises a *ROS Topic*. Once that topic of communication is advertised, a *node* needs to subscribe to it to publish messages and, finally, those messages can be listened by any *node* also subscribed to the same topic. [figura ilustrativa deste processo] Such messages can be of any kind, from simple strings of characters to visual and laser data. It is easy to understand that this information exchanging structure has a great potential when applied to complex multi-sensor systems, since an uniform and standardized communication between sensors facilitates the development of robotic applications.

Another important feature that ROS provides is the possibility to record data logs (*rosbags)* that can be replayed later. This allows for data to be collected in real scenery, to be later treated in a laboratory environment as if it was on the field.

Recently, to standardize the work developed for the ATLAS project, an effort has been made to migrate every application built to work in ROS environment and, for this reason, the work presented in this document was also developed in ROS.

**2.2 – OpenCV**

OpenCV is an popular open source computer vision library written in C and C++ that was designed for high computational efficiency and with a strong focus on real-time image processing applications. It provides an infrastructure to help people building fairly sophisticated vision applications, and contains hundreds of functions that span many areas of application, like factory product inspection, security, medical imaging, robotics and many others. In this work many useful OpenCV functions are used for several purposes. Computing gradients, converting images between colors spaces, resizing images or selecting sub-windows from images are just a few examples of operations provided by OpenCV that are absolutely necessary in this work.

**2.3 – INRIA Dataset**

In order to develop detection applications of objects in complex scenes, where segmentation and other similar approaches are not an option, the use of a learning machine algorithm is absolutely necessary. In a nutshell, these algorithms work by exhaustive learning of positive and negative instances of the problem in question, and once the learning process is finished, they are able to predict on new unseen data. So, in order to develop a detection application, the developer must possess set of positive examples of the object he wants to detect, and, to ensure that the learning engine is correctly taught, the number of positive examples usually needs to be large. Acquiring hundreds, sometimes thousands, of positive instances of an object is obviously a slow and time-consuming task, even more when the data needs to be treated and labeled in laboratory. Fortunately, a handful of pedestrian datasets were made public by the scientific community, and anyone is free to use them.

The INRIA dataset was acquired by (Dalal and Triggs, 2005) with the objective of setting a challenging framework to test the HOG algorithm. Since then, this dataset has been used by most pedestrian detection researchers, as did the authors of *ChnFtr*.

This dataset provides a training set with 1218 images where no pedestrians appear (negative images), and 2416 positive training crops, meaning, pedestrian images cropped from the original scene in which they appear. The fact that the positive examples are already cropped out of the original images largely facilitated the training process. The dataset also provides a different set of images, with 1132 positive crops and 462 negative images to test the detector performance.

This dataset was of utmost importance for the development of this work, as it not only allowed for a facilitated training/testing process, but also allowed for a meaningful comparison between the original algorithm results and the ones achieved in this work. [Colocar imagens exemplo do dataset]

* 1. **– Experimental Setup**

The first step to take in order to carry out the project was the establishment of a platform in ROS for advertising, subscribing, processing and publishing images. For this, two different nodes were created, an image server and an image client.

The image server advertises a *ROS topic,* loads the dataset images from file and publishes it on the topic.

The image client subscribes to the *ROS topic,* and receives the image messages the server sends. Every time an image is published in the topic, a call back function is triggered on the client, and this is when the image is processed using OpenCV tools. Once the image is processed, it is published on another topic for visualization. Image x illustrates this whole process.