A Review of Object Classification Methods for Pedestrian Detection in Autonomous Vehicles

Daniele Gadler
Department of Computer Science,
Technische Universität Kaiserslautern,
67663 Kaiserlautern, Germany
gadler@rhrk.uni-kl.de

***Abstract*—Autonomous driving has been a long-yearned dream since the invention of cars. Nowadays, due to advancements in computer vision, autonomous driving has become a reality, with firms like Tesla and Google showcasing fully-autonomous cars in certain driving scenarios (e.g, on a highway or in a parking). One particular topic has proven to be a real challenge in autonomous driving, namely dynamic pedestrian detection in a moving automated car. Difficulty is given by the fact that algorithms for pedestrian detection need to function in extremely complex scenarios, as pedestrians may be wearing very diverse clothing, may be subject to partial occlusion from objects, carried items or even other pedestrians. In this paper, we provide a review of algorithms for object classification in pedestrian detection, with a focus on autonomous driving. With pedestrian detection, we intend the process of recognizing an object shape as belonging to a pedestrian, whereas with object classification we mean categorizing an object depending on a set of features detected.**

I. Introduction

A. Motivation

AN autonomous car is a vehicle that makes use of computer-assisted driving functions in a certain degree. These functions can range from a vehicle equipped with Continuously Variable Transmission (CVT) to a vehicle that only needs an input destination to reach its destination autonomously. NHTSA, National Highway Traffic Safety Administration, classifies autonomous cars as highlighted in Table I [1]:

In the present analysis, we mainly consider Level 2 and Level 3 self-driving cars as described in Table I .

The benefits of replacing human-driven cars with such self-driving cars have been reported to be mainly the following ones [2]:

- Augmented safety due to reduction of car accidents, both involving pedestrians and cars.
- Reduction in transport time due to improved cars coordination and reduction of car traffic.
- Lower environmental impact, due to reduction of exhaust gas emissions derived by the reduction of traffic.

Considering the fact that drivers appear to account for 94% of car crashes [3], it is sensible to invest and foster research into the field of autonomous driving, in an attempt to replace or aid human beings with computer-based intelligent systems while driving.

TABLE I. NHTSA classification for autonomous driving vehicles

1972 Level 0 Chevrolet Vega	**1998 Level 1 Mercedes S500**	**2016 Level 2 Tesla S**	**2025 Level 3 Uber, Google**	**Level 4 JohnnyCab from Total Recall**
Driver is always in full control of the vehicle.	Driver can regain control or brake more quickly.	Driver shares control as an intermittent operator.	Operator or ride-hailing service cedes full control in certain circuimstances.	The driver selects a destination and doesn't control the car.
Automatic transmission (optional).	Automation of certain control functions (e.g, assisted breaking).	Partial automation of primary control functions working together (e.g: adaptive cruise control)	Steering, throttle, braking and other critical functions are fully automated. Car monitors road conditions and eventually lets the driver re-take control. (e.g: construction)	Fully automated. The car can perform all safety-critical operations and monitor road conditions during the trip. Responsibility for safe operations lies uniquely within the vehicle.

B. Problem Identification

To identify the issues that should be tackled, it is useful to investigate the root critical reasons underlying driver-related crashes.

The most critical reason, accounting for 41% of crashes, appears to be the recognition error, concerning internal and external distractions; the second most critical reason, accounting for 33% of crashes is the decision error, encompassing wrong judgements about other drivers' behaviour or behaving in a wrong manner with respect to the environmental conditions (e.g: driving too fast on a curve). Together, these two types of errors consist of almost $3/4$ of all accidents.

To tackle these highlighted issues, a variety of systems that can interpret sensor information and take adequate decisions to support or aid drivers, can come into play (e.g: radar, GPS and cameras).

In particular, computer vision has been an ongoing field of research in the last few decades and has experienced great advancements: with respect to lasers, cameras provide a cheap and efficient alternative, while at the same time offering a good degree of feasibility. They can operate in the visible spectrum (at daytime), in the infrared one (at nighttime) or even in the thermal one. In many applications, these cameras can also be combined so as to operate in multiple spectra (e.g, in the

thermal infrared)[4].
In this paper, we analyse state-of-the-art algorithms for object classification and pedestrian detection in autonomous driving, which would help reduce the impact of the main critical reason for car crashes: the recognition error.

C. Related Work

This review is mainly based on the Survey for Monocular Pedestrian Detection by M. Enzweiler and D. M. Gavrila [4], which provides a good classification of middle and high-level methods for pedestrian detection, distinguishing between generative and discriminative models. Furthermore, this survey also describes Haar Wavelet-Based Cascade, Neural Networks using Local Receptive Fields (NN/LRF) and Histograms of Oriented Gradients with linear SVM (HOG/linSVM).
A good introduction into Convnets for background-foreground discrimination, along with their performance, is given by Zhang et al. [5].
A review and analysis of Object Classification Methods in ROIs (Regions of Interest) is given by D. Gerónimo et al. [6], where they evaluate learning algorithms (SVM, Adaboost, Neural Networks) as well as middle and high-level approaches. The latter ones are classified by parts-based and holistic approaches, whereas the former ones encompass mainly features onto that high-level approaches are based.
Middle-level features include Haar-like Features, Haar wavelets, edgelets, shapelets, Histograms of Oriented Features, EOH. These algorithms are used by high-level methods such as the Chamfer System, Convolutional Neural Networks.
A general overview of low-level and middle-level feature extraction methods is provided by E. Maggio and A. Cavallaro in their book 'Video Tracking' [7]: for low-level features, analysed topics are Colour, Gradient and Derivatives, whereas middle-level features regard edges and uniform regions.
An introduction into different clustering techniques is provided by L.V. Bijuraj [8]. Finally, an analysis of Deep Learning for object detection in autonomous driving with convolutational neural networks as well as feature extraction is provided by V.V. Gomez [9].
The evaluation of many of these state-of-the-art approaches is carried out by P. Dollár et al. [10], who compare different state-of-the-art approaches on the Caltech dataset and evaluate them.

D. Goal

The goal of this paper is two-fold: on the one hand, we categorize state-of-the-art algorithms for object detection based on which abstraction level they operate (low, medium, high) and point out dependencies between algorithms lying on different layers. On the other hand, we evaluate them and explain which advantages and disadvantages characterize them, as well as their application fields.

II. Analysis of the State of the Art

Following, different algorithms for object classification in pedestrian detection are categorised with respect to the layer on which they operate, their pros and cons and dependencies between algorithms.
We start our categorisation process with *low-level* algorithms (operating on pixels-related features such as colour and pixels' intensity frequency), then inspect *middle-level* algorithms (operating on edges, edgelets, corners, areas of images) and finally analyse *high-level* algorithms (detecting full objects like cars or pedestrians).

A. Low-Level Algorithms for Feature Detection

TABLE II. Main Low-Level Algorithms for Feature Detection

Low-Level Algorithms			
Name	**Advantages**	**Disadvantages**	**Applications**
Color:			
RGB Colour Space Model	Easy to implement.	Device dependent.	Screens, pictures in digital format.
CIELAB Lab Color Space	Device-independent. Can generate all visually perceptible colours.		Interchange format between devices or different colour space models.
YIQ, YUV, YCbCr	Separation of luminance from chrominance.	Often deemed as unintuitive.	Televisions, video and image compression.
HSL,HSV HSB	Separation of hue, saturation and lightness/ brightness.	Unintuitiveness of colour specification. Device dependent.	Image processing, computer vision, colour pickers
Gradient and derivatives:			
Sobel operator	Strong mathematical foundation through first-order derivatives.		Image processing for edge detection.
Laplacian operator	Strong mathematical foundation through 2nd or higher-order derivatives.		Image processing.

1) Colour Space: A colour space is a mathematical representation of our visual perceptions that allows for analysis and management of colours: this a topic of key importance for features recognition, as objects are often characterised by a certain colour that can serve as a distinguishing pattern.

- The *RGB Colour Space Model* is an additive non-linear colour space model based on the three primary colours red, green and blue. Each of these colours is represented on different weighted axes in a three-dimensional cube. These weights are device-dependent and allow for conversion to the device-independent CIE XYZ space, the final resulting colour to display.

- The *Lab Colour Space Model (CIE-LAB)*, in contrast to RGB, is a device-independent colour space model based on three variables: L stands for lightness, A and B respectively for the opposing colours green and red. Though this model could potentially represent all visually perceptible colours, it is generally represented on three axes. This three-axes representation limits the amount of generable colours.

- *YIQ, YUV and YCbCr* are models inspired by the human color responsiveness to colours, as the eye is

more sensitive to changes in the orange-blue band than in the purple-green one. The main characteristic of these systems is the separation of luminance (Y) from chrominance information (respectively I and Q, U and V or Cb and Cr). These formats are employed in different applications, with YIQ being used in NTSC televisions, YUV for analog encoding of colour and YCbCr for digital encoding of colour.

- The *HSL (Hue, Saturation, Lightness), HSV (Hue, Saturation, Value), HSB (Hue, Saturation, Brightness)* colour spaces represent non-linear transforms of RGB, with separation of lightness, hue and saturation from luminance or brightness.
  Being a direct modification of RGB, these colour spaces are all device-dependent. The main difference between them is the following one: S (Saturation) is defined differently in these colour spaces and requires conversion to fit respectively the definition of brightness or lightness. In fact, lightness L (the amount of white in the colour space) is a completely different concept from brightness (amount of light, of any colour). [11]

2) Gradient and Partial Derivatives: Gradient and partial derivatives help identify local spatial intensity changes in a picture. These can be useful to find contours of objects in the background or foreground (for instance, pedestrians on a street), which are essential for object classification and detection. The main operators are:

- The *Sobel Operator* performs a spatial gradient measurement on a picture in order to highlight regions characterised by high intensity values over a small distance. These regions are also known as ”edges”.
  The process of computing edges involves the use of a 3x3 convolution kernel vertically and one horizontally, to produce gradient measurements of the gradient component for the two orientations. The combination of these components will produce the absolute magnitude at each point, which is then output to the user [12].
- An undesirable effect of performing image-gradient operations is the highlighting of high frequency areas in correspondence with sensor-generated noise. To solve this problem, the *Gaussian Filter* helps build robustness to noise by using the least squares estimate obtained from the structure tensor [13].
- The *Laplacian Operator*, analogously to the Sobel Operator, also makes use of the gradient to highlight steep changes in an image by summing second order derivatives on the X and Y axes. Afterwards, an approximation of the Laplacian can be produced by convolving the image with a 3x3 kernel. Since the Laplacian Operator is very sensitive to noise, the Laplacian can be combined with a Gaussian filter, generating an LoG operator.

B. Middle-Level Feature Detection

1) Local intensity-based models: Local intensity-based models extract features from an image based on the processing of pixels' intensity in ROIs (Regions of Interest, a particular rectangular region within an image) or points of interest within it, generally to compute variations between different ROIs. Following, different processing techniques are laid out:

TABLE III. MAIN ALGORITHMS FOR MIDDLE-LEVEL FEATURE DETECTION

Middle-Level Algorithms				
Name	**Advantages**	**Disadvantages**	**Application**	**Dependency**
Local intensity-based models:				
Moravec Corner Detector	Not computationally demanding.	Non-directional independent output. Non-repeatible algorithm.	Object recognition. Image mosaicing.	Interest points
Harris Corner Detector	Improvement of Moravec's algorithm. Repeatability in detection of interest points.		Object recognition, Image mosaicing.	Interest regions
Codebook features	Effective detection of feature patches.		Extracting feature vectors.	Interesting points
Haar-Like Features	With respect to pixels-based features, it is computationally inexpensive.		Categorize subsections of an image.	Rectangular regions of interest
Local Edge Structure				
HOG - Histograms of Oriented Gradients	Invariant geometric and photometric transformations. Optimal when pedestrian in upright position.		Object detection in image processing.	Intensity gradients. Edge directions.
SIFT - Scale Invariant Feature Transform	Invariant to uniform scaling and orientation. Robust to clutter and partial occlusion.		Detect and describe local features of an image.	Interesting points.
Clustering				
Partitional clustering	Efficiency and speed in building clusters.	Does not consider objects' proximity.	Pattern recognition.	
Hierarchical Clustering	Better built model accuracy.	Computationally more expensive than partitional clustering.	Pattern recognition.	

- The *Moravec Corner Detector* computes intensity variations in eight different directions of a rectangular region constructed around a central pixel. This allows to find regions with large intensity variations in multiple directions by computing the sum of the squared differences between the pixel values and thresholding the resulting value to find local maxima in the image. However, since the output is anisotropic, the output interest-points are not rotation-invariant, and this hinders repeatability of this procedure.
- The *Harris Corner Detector* improves over the Moravec Corner Detector, allowing for procedure's repeatability in the detection of interest points, not taking into account the target orientation. It does so by making use of a weighted version of the differential score and a Taylor

approximation.

- A codebook is a manual containing correspondences between codes and their equivalent value in an image (e.g: a certain area corresponds to a body part). Through *Codebook Features*, we intend a codebook of distinctive features for pedestrians, inter-connected with each other by geometrical relations.
  This codebook can be created from training on sample data to provide a trained model of the considered pedestrian class (a "bag-of-parts", along with their relations). This model may then serve for detecting further pedestrian occurrences in images. [14].
- *Haar-Like Features* offer an alternative, inexpensive way to process image pixel intensities. Instead of processing the RGB value of a region pixel-by-pixel, a Haar-like feature considers different smaller adjacent rectangular regions (sub-regions) contained in a bigger "window" within an image. Afterwards, it computes the sum of pixel intensities for all sub-regions within the identified "window" and calculates the difference between the considered sub-regions: this difference can then be used to categorise subsections of an image. [15]

2) Local Edge Structure: Research has shown the efficiency of considering the local edge structure of a region instead of pixel-intensity for human detection [16] and the detection of image features from scale-invariant keypoints (SIFT) [17]. Main methods that make use of local edge detection are listed below:

- *HOG, Histograms of Oriented Features*. HOG subdivides an image into small connected regions (cells) and computes an histogram of gradient directions for pixels in each cell. The edge directions, along with the intensity gradient for every cell, are then concatenated and can provide a good representation of an image.
  This representation can be further improved to resist to changing illumination and shadowing by normalising the contrast for all cells. This normalisation can be carried out by computing a measure of the intensity over a larger region of the image (a block) and applying this value onto the contrast of all histograms.
- *SIFT, Scale Invariant Feature Transform* allows to identify interest points from a set of reference images and store them in a database, analougously to a database of trained data. Afterwards, features extracted in new images can be compared with the features stored in the trained database features, computing the euclidean distance from one other and finding most similar models for them.
- *EOH, Edge Orientation Histograms* help classify ROIs in an image. Firstly, they compute the gradient magnitude of an image, then distribute pixels into several different bins according to gradient orientation.
  Features are defined as the ratio between the two summed gradient magnitudes of an ROI. [18].

3) Clustering: Clustering consists of grouping together sets of objects such that objects contained in the same group (a cluster) are more similar to each other than the ones contained in other clusters. Different techniques for building such clusters are presently outlined:

- *Partitional Clustering*: Firstly, the arithmetic mean of every object's attribute vectors is computed. The resulting value is then used to assign an object to a cluster, which will contain objects with similar resulting values.
- *Hierarchical Clustering*: In this approach, objects are grouped together with other objects located around them than farther ones, based on the assumption that objects are more related to the ones around them. Afterwards, the resulting clusters are grouped again with other clusters lying near them.

C. High-Level Feature Detection

1) Generative Models: Generative models are based on the Bayesian conditional density function, which computes the posterior probability of a shape and classifies it as belonging to a pedestrian or not. Main methods using this approach are the following ones:

- *Continuous Shape Cue Model*: A model for the class-conditional density of pedestrians can be extrapolated from a training dataset, which can then be represented in linear or non-linear spaces to preserve physically plausible regions. A drawback of this approach is the increasing amount of data required to train the model in a non-linear space, which goes hand-in-hand with the increasing complexity of the model.
- *Discrete Shape Cue Model*: A pedestrian shape is represented through a discrete set of exemplar shapes. An advantage of this approach is the high specificity of matched shape models (which are highly and strictly characteristic of the analysed one), yet a drawback is represented by the wide amount of shapes that need to be stored and matched against the considered one.
- *Markov Field Layer*: This approach makes use of a two-layer statistical model that characterises shape variations by representing shapes through a distributed connected model, based on Bayes priors. A Hidden Markov Field layer is employed to find these prior probabilities, associating the likelihood of image observations. The benefits of this approach are manifold, as the generated model's robustness to partial shape occlusions and background clutter is increased [19].

2) Classifiers: Classifiers are algorithms that identify an object as belonging to a certain category, depending on their features. After being trained with data and identifying features' patterns within them, such algorithms are then able to apply discovered patterns to new data. These algorithms are often referred to as machine learning algorithms and following, the main ones for image processing are listed.

- *SVM - Support Vector Machine* is one of the most popular machine learning algorithms for pattern classification in image processing. This procedure maximises the margin of a linear decision boundary to obtain the greatest separation possible between the considered object classes. To detect pedestrians' features, SVM can be

used with non-linear feature sets (characterising human beings). Furthermore, non-linear feature sets operating on higher or infinite dimensional spaces can on one hand lead to a performance increase, which is paid in terms of an increased memory space and computational cost.

- *Adaboost* is also a very widely used machine learning algorithm: it is used to build strong classifiers by combining weighted weak classifiers for a single feature. A main advancement has been brought forward by Viola et al. [20], who introduced cascades of motion and appearance in pedestrian detection.
  In each layer, Adaboost iteratively constructs a strong classifier, basing itself on errors made on previous layers, hence generating increasingly complex detectors for every layer.
  This approach has proven to be particularly efficient at distinguishing non-pedestrians from pedestrians in an image.
- *HOG/ linSVM, Histograms of Oriented Gradients with linear SVM*:

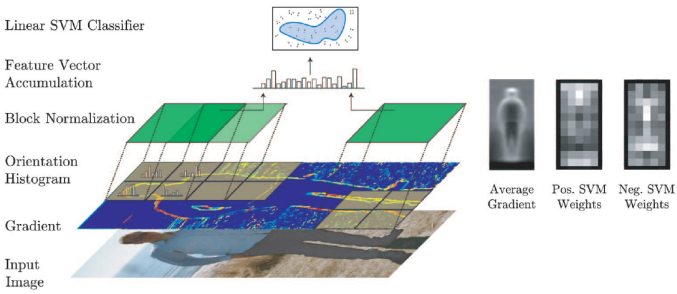

Fig. 1. Graphical Representation of an HOG / linear SVM classifier [4].

This classifier builds a feature vector that is characteristic of an analysed human shape through a multi-layer approach.
Firstly, local shapes are modelled using normalised Histograms of Oriented Gradients (HOG). Afterwards, gradients are categorised depending on their orientation and weighted by their magnitude in a spatial grid of cells with overlapping blocks contrast normalisation. In each overlapping block, we extract a feature vector by considering the histograms resulting from the cells. The final classification vector is the concatenation of all the resulting feature vectors.

- *Logistic Multiple Instance Learning* represents a recently developed supervised machine-learning algorithm that is particularly efficient with incomplete knowledge of training labels.
  An example by Babenko [21] explains how LMIL solves this incompleteness problem:

TABLE IV. MAIN HIGH-LEVEL ALGORITHMS FOR FEATURES DETECTION

High-Level Algorithms				
Name	**Advantages**	**Disadvantages**	**Application**	**Dependency**
Generative Models				
Continuous Shape cue model	Reducing variations in pedestrian appearance.	Training data quantity increasing with model complexity.	Pedestrian shape recognition.	Bayes theorem.
Discrete Exemplar-based Shape Cue Model	Highly specificity of model.	Requires large amount of training data.	Pedestrian shape recognition.	Bayes theorem.
Markov Field Layer	Increased robustness to partial occlusion in shape.		Pedestrian shape recognition.	Hidden Markov Model. Bayes theorem.
Deep Learning	Multiple layers processing with linear and non-linear processing.	Requires large amounts of data. Extreme computational complexity.	Object classification and detection.	Hierarchical clustering
Neural Networks	Little statistical training. They can detect complex non-linear relations.	Great computational complexity. Tendency to overfitting.	Feature detection.	Animal brain's structure.
Classifiers				
SVM - Support Vector Machine	Powerful pedestrian classification.	Higher computational cost and memory requirements if used on a higher dimensional space, compensated by performance.	Pedestrians pattern classifi-cation.	Hyperplane, linear decision boundary.
Adaboost	High processing speed. It can quickly distinguishing non-pedestrians from pedestrians.		Building strong classifiers for distingui-shing pedestrians from non-pedestrians.	Cascade layers.
HOG / linSVM Histograms of Oriented Gradients with linear SVM	Classification feature vector.		Classificaton of pedestrian features.	SVM and HOG.
Logistic Multiple Instance Learning	Faster detection than HOG / linSVM and Adaboost.		Pedestrian Shape Recognition.	Cascade layers.
Non-adaptive Features				
Haar Wavelet Cascade	Efficienct detection of pedestrians.		Classifier for pedestrian detection.	Decision tree based on detector layers. Adaboost.
Adaptive Features				
NN/LRF Neural Network using Local Repective Fields	Powerful features detection in pedestrian detection.		Detection of pedestrians with varying features.	Neural networks.
Others:				
Implicit Shape Model	Small training set thanks to its flexible features representation.		Detecting objects in cluttered real-world scenes.	Hessian-Laplace keypoints detector. Hough Voting. Chamfer Distance.
Chamfer System	High-Performance and more efficient computation over a brute-force approach.		Pedestrian shape identification, objects detection for moving vehicles.	Hierarchical clustering.

Imagine several people, and each of them has a key chain that contains few keys. Some of these people are able to enter a certain room, and some are not. The task is then to predict whether a certain key or a certain key chain can get you into that room. To solve this problem we need to find the exact key that is common for all the positive key chains. If we can correctly identify this key, we can also correctly classify an entire key chain - positive if it contains the required key, or negative if it does not.

- *Deep Learning*: learns abstract high-level structures present in a training dataset, allowing for hierarchical feature extraction. Deep Learning proposes a brand new approach to classification, based on the brain's neurons functioning. Recent advancements in this field have sparkled great interest in the usage of Deep Learning since the 90s and this innovative machine learning algorithm has produced accurate results never reached before. However, a downside of Deep Learning is represented by its extreme computational complexity and resource demand.
- *Neural Networks* share many similarities with Deep Learning, as they are also based on the brain's neurons functioning. With Neural Networks, we usually refer to *Feed-forward Neural Networks*, where the input just moves in one direction (forward), from the input layer of neurons onto the next neuronal layer, until it is output at the last layer's level. Neural Networks containing cycles among the units are called *"Recurrent Neural Networks"*, whereas Neural Networks featuring multiple layers are called *Deep Neural Networks*.
  Neuronal Networks are mainly based on two types of perceptrons (algorithms for the classification of input).
  1) Single-Layer Perceptrons: In this structure, inputs nodes are passed to output nodes via a series of weights, so as to produce a single layer of output nodes. In each node (a neuron), the sum of the products of the weights and the inputs is calculated. If the resulting value is above some threshold (usually 0), the neuron is activated and takes the computed value (typically 1). Otherwise, it takes the deactivated value (typically -1).
  2) Multi-Layer Perceptrons: Each neuron in one layer has directed connections to the neurons of the subsequent layer. One of the most renowned techniques for training neurons is back-propagation: output values in different layers are compared with the correct answer to a certain error-function. The error is then passed again through the network and the algorithm adjusts the weights of each neural connection, reducing the error function by a small amount, until it converges to a certain state with small error.

  Neural Networks are able to learn multi-layer invariant features that characterise objects through proper training. [22] For this reason, they find applications in Object Recognition and their classification. However, Neural Networks share the resource and computational complexity of Deep Learning, as they need to be training with vast amounts of data to generate very complex models.

3) Non-Adaptive Features: They represent features that are invariant to training data and are used as a reference for features in an analysed image. Presently, one of the most widely used approaches is presented:

- *Haar Wavelet-Based Cascade*: In this approach, images are split into layers and a decision tree builds an ever-increasing complex detector on every layer through a machine learning algorithm (e.g: Adaboost). Namely, Adaboost constructs a classifier based on a weighted linear combination of features, which are characterised by the lowest error on the training set.

4) Adaptive Features: They consist of ever-changing features that adapt to the training sample used.

- *NN/LRF Neural Network using Local Repective Fields*: This approach uses neurons' branches, with each neuron in every branch receiving input from a limited region in the input layer (the receptive field) as a spatial feature detector. The output consists of two neurons, where each of them represents the posterior probability of a shape, identifying it as belonging or not belonging to a pedestrian class.

5) Others:

- *Implicit Shape Model*: avoids Regions of Interest generation by using a Hessian-Laplace keypoints detector, which computes a shape context descriptor for every keypoint. All of these keypoints are then clustered to construct a codebook of features.
  Afterwards, during the detection phase, all keypoints use Hough Voting to elect an object hypothesis. In this manner, the candidate generation step can be omitted. Fine pedestrian silhouette segmentation is provided through the Chamfer Distance.
- *The Chamfer System*: provides an efficient solution to shape-based object detection. For this reason, the Chamfer System has been used for detecting pedestrian silhouettes. A flexible and adaptable model is generated through training on a dataset and there, its pixel-based correlation approach eliminates the need for error-prone contour segmentation. It also features a hierarchical clustering approach (in both shape and transformation space), resulting in consistent performance gains over brute-force approaches. [23].

III. Conclusions

In the last two decades, considerable advancements have been made in the field of Object Detection for Autonomous Driving, as reported in the Literature surveys by Enzweiler et al. [4] and Geronimo et al. [6].
As far as middle-level features are concerned (not considering classifiers), Geronimo et al. consider HOGs and shape-based appearance as state-of-the-art features, with appearance-based

methods representing the future direction of research, both for pedestrian and object detection. Haar-Like Features also find extensive usage in Haar-Wavelet based cascades and Hierarchical Clustering is a core part of different other techniques (e.g: the Chamfer System).
Other methods are also dependent on appearance-based procedures: for example, the Chamfer System cannot operate solely by itself, but requires an extra appearance-based step, represented by the computation of pixel-based correlation.
To establish the superiority of a high-level method with respect to another one, Enzweiler et al. perform benchmarking [4] of HOG/linSVM , NN/LRF and Haar-Wavelet Cascade both in 2D and 3D scenarios.
In the 2D scenario, HOG/linSVM outperforms NN/LRF and Haar Wavelet Cascade with a value of 0.045 false positives against 0.38 and 0.86 respectively for NN/LRF and Haar Wavelet Cascade, as shown in Figure 2.

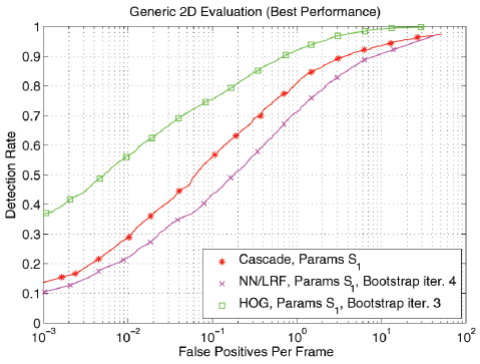

Fig. 2. Resulting evaluation of Haar-Wavelet Cascade, NN/LRF and HOG/linSVM in a 2D benchmark [4]

When combined with temporal tracking, HOG/linSVM performs with an even higher level of precision.
In a real-time 3D pedestrian-detection scenario, HOG/linSVM, NN/LRF, Haar-Wavelet Cascade and Shape-Texture detection are compared with a time constraint of 250 ms and 2.5s per frame. The results are shown in Figures 4 and 5.

- *250ms*: Haar-Wavelet based Cascade outperforms the other approaches with little processing time, whereas Shape-Texture Detector and HOG/linSVM behave similarly.
- *2.5s*: HOG/linSVM benefits from the increased processing time, performing similarly to Haar-Wavelet based Cascade.

These results highlight the fact that state-of-the-art detectors are still far from reaching perfection. To fill this gap, Enzweiler et al. suggest that a pre-processing stage could be incorporated, so as to reduce the image search space, making use of other cues such as motion or depth in the processing phase.
In fact, Object classification is just one step in the overall process of Pedestrian Pedestrian. As highlighted in Figure 3, object classification is dependent on the preceding step (foreground segmentation) and provides data for the succeeding one (verification/refinement). The overall process, incorporating several different phases of processing, will yield better results than a mere classification.

Performance improvements could also be achieved via better classification methods, such as local receptive fields or gradient histograms. Furthermore, an increase in training data could but be beneficial for the performance of algorithms.
Finally, advancements in machine learning algorithms will certainly pave the way to even faster and more performing detectors, as Adaboost and SVMs could prove. Currently, the computational requirements of Neural Networks and Deep Learning represent an obstacle for their application in real-time object detection. Yet, Deep Learning and Neural Network are certainly the future of pedestrian detection in autonomous driving and could help fill the gap still existing in current state-of-the-art detectors.

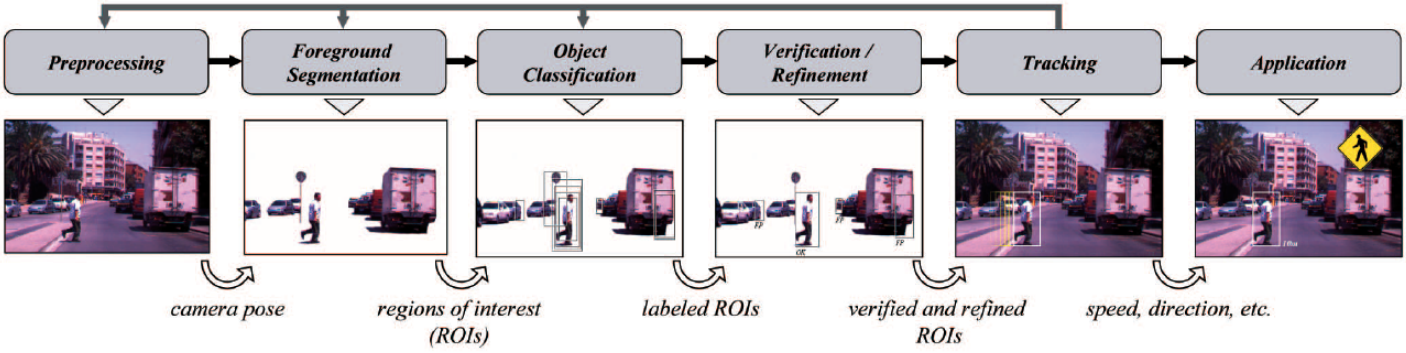

Fig. 3. Pedestrian Detection's steps showcased. In our analysis, we mainly on Object Classification [6]

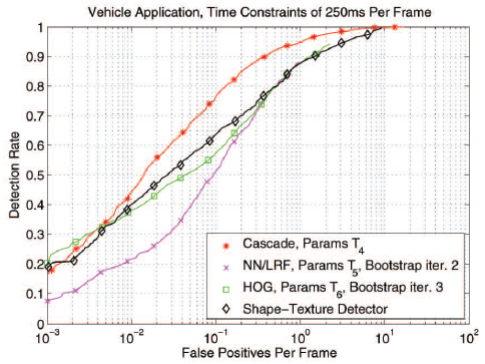

Fig. 4. HOG/linSVM, NN/LRF, Haar-Wavelet Cascade and Shape-Texture detection in 3D real-time pedestrian detection scenario with a time constraint of 250ms [4]

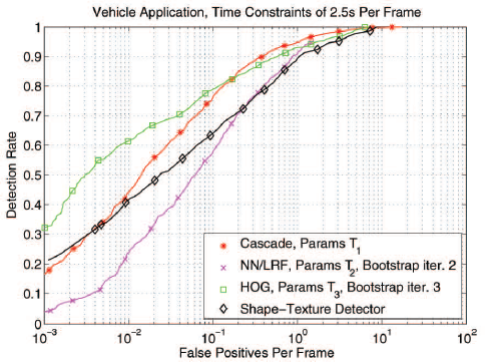

Fig. 5. HOG/linSVM, NN/LRF, Haar-Wavelet Cascade and Shape-Texture detection in 3D real-time pedestrian detection scenario with a time constraint of 2.5s [4]

ACKNOWLEDGMENT

The author would like to thank Tanittha Sutjaritvorakul for shading light onto the world of computer vision in autonomous driving and providing indispensable help and guidance as my supervisor for this Seminar in Embedded Systems and Robotics.

REFERENCES

[1] "Uber's First Self-Driving Fleet Arrives in Pittsburgh This Month," https://www.bloomberg.com/news/features/2016-08-18/uber-s-first-self-driving-fleet-arrives-in-pittsburgh-this-month-is06r7on, [Accessed on 16/1/2017].

[2] D. J. Fagnant, K. M. Kockelman, "Preparing a nation for autonomous vehicles. opportunities, barriers and policy recommendations," October 2013, [Accessed on 1/1/2017].

[3] "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," February 2015, [Accessed on 1/1/2017].

[4] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, 2009, pp. 2179–2195.

[5] S. Zhang, R. Benenson, M. Omran, J. H. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" *CoRR*, vol. abs/1602.01237, 2016. [Online]. Available: http://arxiv.org/abs/1602.01237

[6] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, July 2010.

[7] E. Maggio and A. Cavallaro, *Video Tracking: Theory and Practice*, 2010.

[8] L. Bijuraj, "Clustering and its applications," in *Proceedings of National Conference on New Horizons in IT-NCNHIT*, 2013, p. 169.

[9] V. V. Gomez, "Object detection for autonomous driving using deep learning," Ph.D. dissertation, 2015.

[10] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, April 2012.

[11] "HSL, HSB and HSV color: differences and conversion," http://codeitdown.com/hsl-hsb-hsv-color/, [Accessed on 16/1/2017].

[12] "Sobel Edge Detector," http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm, [Accessed on 16/1/2017].

[13] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, Jun 1998, pp. 232–237.

[14] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1475–1490, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2004.108

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[18] K. Levi and Y. Weiss, "Learning object detection from a small number of examples: the importance of good features," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, June 2004, pp. II–53–II–60 Vol.2.

[19] Y. Wu and T. Yu, "A field model for human detection and tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 753–765, May 2006.

[20] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct 2003, pp. 734–741 vol.2.

[21] B. Babenko, "Multiple instance learning: algorithms and applications," *View Article PubMed/NCBI Google Scholar*, 2008.

[22] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 253–256.

[23] "The Chamfer System," http://www.gavrila.net/Research/Chamfer_System/chamfer_system.html, [Accessed on 19/1/2017].