A Context Aware and Video-Based Risk Descriptor for Cyclists

Miguel Costa[1] and Beatriz Quintino Ferreira[2] and Manuel Marques[3]

***Abstract*—Aiming to reduce pollutant emissions, bicycles are regaining popularity specially in urban areas. However, the number of cyclists' fatalities is not showing the same decreasing trend as the other traffic groups. Hence, monitoring cyclists' data appears as a keystone to foster urban cyclists' safety by helping urban planners to design safer cyclist routes. In this work, we propose a fully image-based framework to assess the route risk from the cyclist perspective. From smartphone sequences of images, this generic framework is able to automatically identify events considering different risk criteria based on the cyclist's motion and object detection. Moreover, since it is entirely based on images, our method provides context on the situation and is independent from the expertise level of the cyclist. Additionally, we build on an existing platform and introduce several improvements on its mobile app to acquire smartphone sensor data, including video. From the inertial sensor data, we automatically detect the route segments performed by bicycle, applying behavior analysis techniques. We test our methods on real data, attaining very promising results in terms of risk classification, according to two different criteria, and behavior analysis accuracy.**

I. Introduction

Bicycles are winning back importance in our society as a sustainable means of transportation, specially in urban areas [1], [2]. In fact, not only do they bring positive impact to the environment, but also to public health and traffic [3]. Both the European Union and the USA are committed to raise the number of cyclists while increasing cycling safety [4]. Notwithstanding, we have witnessed a much lower decrease (3%) in the number of cyclist fatalities when compared to the fatalities reduction in the other traffic groups (around 18%) [5].

Collecting traffic data, in particular cyclists' data, is very important for urban planners and a keystone to design safer cycling routes.

With the advent of smartphones and other mobile wearable devices, acquiring massive sensory data for behavior analysis has become not only highly affordable but also a common practice [6]; so these appear as a perfect match to this task.

In this work we explore this synergy: use sensor data to assess the route risk, fostering safety and mobility for urban cyclists.

In fact, several recent studies have been focusing on collecting and analyzing different types of traffic data (video, GPS, acceleration, orientation) in an attempt to evaluate and improve road operation and safety regarding cyclists [2], [7], [8], as well as promote active commuting, which leads to reduction of air pollution and congestion in traffic networks [9].

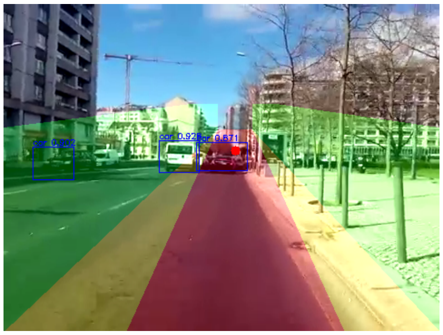

Fig. 1: Risk analysis descriptor: The estimation of the Focus of Expansion (red point) enables to define three risk zones (red, yellow and green); our risk descriptor is correlated with the occupancy of each zone by detected objects (blue rectangles). In this image, the detection of a car in the red zone (cyclist' route) indicates a possible high risk situation. Differently, the car in the green zone represents a lower risk for the cyclist.

In particular, we believe that monitoring cyclists route risk can be valuable to improve safety as well as help guide city planning. Our previous work, the SMARTcycling tool presented in [10], was the first to automatically identify generic driving events that may condition cyclists' real commuting experience. In the latter work, stressful events were detected using a bio-metric sensor. Although the method was able to assign stress levels to segments of the paths performed by the cyclists, it required a posteriori visual inspection of the acquired images in order to understand what particular event generated the stress level variation (e.g. other passing by cyclists, cars or pedestrians, road anomalies, to name a few). Moreover, it was observed that similar patterns were obtained during stress and effort situations (e.g. due to terrain elevation), requiring disambiguation through image and GPS data analysis. We may also postulate that the identification of stressful events based on biological signals may be user-dependent, varying accordingly to the cyclist's experience, comfort, or physiological characteristics, among other factors.

We center our approach on smartphone data, taking advantage of the video captured by our mobile application (dubbed Bike Monitor) to develop an alternative method based on optical flow and focus of expansion (FOE) to assess risky events from the external factors from the route, providing also

[1] Miguel Costa is a Master student with the ECE department, Instituto Superior Tecnico, 1049 Lisboa, Portugal miguel.n.costa@tecnico.ulisboa.pt

[2] Beatriz Quintino Ferreira is a Ph.D student with the ECE department, Instituto Superior Tecnico, 1049 Lisboa, Portugal beatriz.quintino@tecnico.ulisboa.pt

[3] Manuel Marques is a Researcher with the ECE department, Instituto Superior Tecnico, 1049 Lisboa, Portugal manuel@isr.ist.utl.pt

the context for each situation (see Figure 1).

Therefore, we build on the SMARTcycling tool from [10], proposing a novel fully image-based method to assess the cyclist's route risk, which is also context and motion aware. Additionally, we introduce significant improvements in the Bike Monitor app towards a more exhaustive and reliable data acquisition, including performing behavior analysis so that we only analyze the segments of the route in which the user is actually riding a bicycle (as opposed to walking or riding a motorized vehicle). Furthermore, relying solely on the cyclist's smartphone image and sensor data is a step towards a cyclist invariant method.

To the best of our knowledge, this is the first automatic method to identify, contextualize (using images), and assess dangerous riding events for cyclists entirely based on smartphone data. More information about the SMARTcycling project and the Bike Monitor app can be found at: http://users.isr.ist.utl.pt/~manuel/smartbike/.

We highlight the following contributions of our work:

- Image-based and context-aware assessment framework of dangerous events (based on semantic and optical flow descriptors), described in Section IV;
- Behavior analysis based on smartphone sensor data, automatically delimiting the portions of the path performed riding a bicycle, as described in Section V.

Moreover, and preceding the previous contributions, we introduce improvements on the SMARTcycling tool. Specifically, we develop new features for the Bike Monitor app, mainly at the back-end layer, but also: user profile registration, video acquisition/upload from the smartphone camera, report and registration of performed routes in a map allowing post inspection. This is described in Section III;

II. Related Work

In the past few years, sensing human activity has become ubiquitous and traffic has been no exception. In this vein, several studies have focused on collecting traffic data to monitor road conditions [11], [12], roadway operation [8], [13], and assessing driving experience [14], [15], [16]. However, the large majority of these works [11], [15], [14], [16], [12] target motorized vehicles as these are still dominant in today's traffic volume. Nevertheless, bicycle usage has recently grown mainly in urban areas [2]; and perhaps the marginal decrease of cyclists' fatality in comparison to all other road groups [5] is a by-product of this trend. These facts have raised awareness to cyclists' safety, and the research community is starting to give more and more attention to this issue (see, for example, the February 2017 Safety Science special issue on Cycling Safety).

Compared to motorized vehicles, collecting and processing cyclist data is more challenging, as bicycles are less stable (no suspension) resulting in noisier data. This lack of stability is specially problematic when processing images acquired by a smartphone attached either to the bicycle or the cyclist.

Cara et al. [7] circumvent this issue by using an instrumented car to acquire data in order to classify car-cyclist scenarios. In this work the authors test machine learning algorithms on bicycle-car interaction data to classify safety-critical scenarios, envisaging the development of Advanced Driver Assistance Systems (ADAS) that support cyclist protection.

Despite the aforementioned technical hurdles, some recent studies address cycling experience by equipping bicycles with on-board sensors. Aiming at specialized cycling intelligent systems, [17] proposes a framework to understand bicycle dynamics and cyclist behavior. Such framework and collected data can be seen as an important pre-requisite to the development of bicycle suited applications.

Concerning cyclists' safety, [18] and [19] study the relation between the number of cyclists going through a given lane or intersection and the risk of crash with other cyclist and motorist, respectively. More recently, Strauss et al. [2] use a large sample of GPS cyclists' trip data acquired via a smartphone application in order to validate deceleration rate as a surrogate safety measure. Particularly, the authors explore the correlation of deceleration with accidents at intersections as a potential proactive measure to prevent cyclist injuries.

Yet, in terms of the sensors used, there has been practically no distinction between assessing drivers' or cyclists' experience, as previous methods usually depend on inertial sensor data (such as accelerometer or gyroscope).

In [10] we introduced a new approach to detect and identify driving events primarily based on processing images from an action camera. We are able to overcome the issues of using a camera mounted on the bicycle as an acquisition sensor, since the natural shake of the cyclists movement is filtered at the computation of the optical flow. In its previous version, the SMARTcycling tool captured and processed data from the cyclist's smartphone, an action camera, and a cardio acquisition belt. Applying image processing techniques based on optical flow descriptors to the action camera videos, the SMARTcycling tool showed good accuracy on driving events classification and road condition identification. This tool was also able to evaluate cyclists' stress using the ECG data collected from a bio-metric belt.

Due to its amenable properties, in terms of set-up and data acquisition, we claimed that SMARTcycling [10] paves the way to large scale assessment, as cities often provide public bicycle sharing programs, where it can be easily deployed.

In this work we delve into more involved computer vision and image processing techniques to be able to automatically identify and contextualize dangerous events from external factors, sparing both the action camera and the bio-metric belt, which imply a more complex set-up. The descriptor used in [10] was context independent (splitting the image into fixed zones), we now use a different and richer approach that encodes the context surrounding the cyclist when performing event detection. Moreover, contrary to our previous work, we analyze the whole image, incorporating motion, temporal dependence and image semantics.

Regarding semantics, Aly et al. [20] propose an approach to crowd-sense users' smartphones to automatically enrich digital maps with semantic road information such as road condition, bridges or crosswalks. However, and once again, the proposed algorithms only rely on inertial sensor measurements.

Here we follow a different direction, taking advantage of the good properties yielded by using images as primary source of data. Indeed, computer vision techniques have been applied, for quite some time, to traditional cyclist monitoring tasks as volume counts [21] and average speed, due to their reliability and efficiency when compared to manual methods [22]. However, so far no work has addressed identification of dangerous situations using an on-board smartphone camera.

We apply state-of-the-art classification methods (convolutional Neural Networks, specifically the Faster R-CNN from [23]) to obtain the localization and presence probability of objects in the image. The semantics provided by object detection and classification allows to interpret and understand the detected dangerous situations, providing much more insight than other types of measurements.

III. Bike Monitor App

As introduced in [10], the SMARTcycling tool has its own smartphone data acquisition interface - the BikeMonitor app.

Bike Monitor runs on Android operative system and has a very simple and intuitive interface that allows user profile registration, start and stop data recording, and upload the recorded data to the server.

Upon registration, the user is asked to provide her/his age, gender, cycling experience level and bicycle characteristics (suspension/no suspension). This data is stored in the server and since it is organized by user account the profile is automatically associated with new uploads from the same user.

In this new version, we further explore the rich sensing capabilities of today's smartphones, adding the recording of the following signals: speed (from GPS), linear and gravitational acceleration along the three axes (X,Y,Z), rotation matrix, orientation, and GPS uncertainty. The interval between acquisitions is now 0.1s (was 0.5s), and all signals are indexed by a time-stamp, allowing a time synchronized processing.

In addition to inertial sensors and GPS data, the Bike Monitor app has now the option to record video and sound from the smartphone camera and microphone, respectively. Bearing in mind battery life issues, video acquisition is configurable in terms of quality (low or high) and frequency (1, 5 or 30 fps).

After uploading the recordings from each journey, the user receives an automatically generated map summarizing the ride.

IV. Image-based risk assessment

In order to provide a framework to detect and assess risky events for cyclists, we use video sequences from the smartphone camera and combine different computer vision techniques to obtain descriptors based on optical flow and semantics. In particular, we start by estimating the FOE to embed the cyclist motion into the descriptor. We then compute a risk descriptor that considers the objects present in the image and the division into zones according to the estimated FOE. Finally, we can assess risk considering different criteria, by using the obtained descriptor and computing a specific distance metric for the specified criteria.

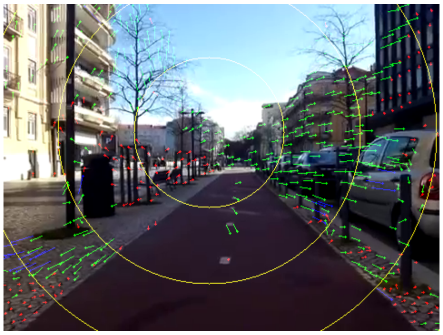

Fig. 2: Optical Flow vectors and associated weights: red corresponds to $m_i = 0.1$, blue $m_i = 0.75$ and green $m_i = 1$.

a) Estimating the FOE: Differently from [10] and before computing the optical flow, we split the image into 16 zones and filter each zone with the histogram equalization method (CLAHE) [24] to enhance contrast and edges definition. We apply the Shi-Tomasi corner detector [25] and the feature extraction method from [26] to find sufficient and evenly distributed points of interest, even in regions with low texture.

The optical flow vector v_i on the image point p_i is computed applying the Lucas and Kanade algorithm [27]. With the optical flow vectors, our goal is to compute the focus of expansion (FOE): a single point in the scene where all the velocity vectors meet. To improve robustness to outliers, we incorporate spatio-temporal prior knowledge about the optical flow and FOE in our 2D images. As Figure 2 shows, the magnitude of the optical flow vectors increases with the distance to the FOE [28] and we exploit this fact to iteratively perform outlier rejection. According to Figure 2, we first divide the image into 4 concentric circles centered around the previous calculated FOE. We then calculate the distribution of the optical flow vectors magnitude in each annulus (formed by the circle excluding its inner circles) and on the innermost circle. Given the average magnitude of its zone $\overline{v_{\Omega_i}}$, each optical flow vector v_i has an associated magnitude weight m_i, according to the following expression:

$$m_i = \begin{cases} 0.10 \text{ , if} & \text{abs}(\|v_i\| - \overline{v_{\Omega_i}}) \geq (\overline{v_{\Omega_i}})^{\frac{2}{3}} \\ 0.75 \text{ , if} & (\overline{v_{\Omega_i}})^{\frac{1}{2}} < \text{abs}(\|v_i\| - \overline{v_{\Omega_i}}) < (\overline{v_{\Omega_i}})^{\frac{2}{3}} \\ 1.00 \text{ , if} & \text{abs}(\|v_i\| - \overline{v_{\Omega_i}}) \leq (\overline{v_{\Omega_i}})^{\frac{1}{2}} \end{cases} , \tag{1}$$

where $\overline{v_{\Omega_i}} = \frac{\sum_{j \in \Omega_i} \|v_j\|}{|\Omega_i|}$, Ω_i is the set of indices whose optical flow vectors v_j are in the same annulus of v_i and abs$(a) = |a|$.

The previous process is only valid for static scenes. However, this is not the case for traffic images, as though their background is static there are objects moving. In order to discover the non-static points, we feed the Faster R-CNN [23] with each image frame to find the class and location (given as a bounding box) of the objects present. Since the probability that the detected objects are static is low (our classes of interest are persons, bicycles and ground motorized vehicles), we weight the flow vectors associated with each object by the negative exponential of the confidence score s output by the neural network. Equation (2) shows the weight o_i assigned for each optical flow vector v_i[1].

$$o_i = e^{-s_i}. \tag{2}$$

This way we minimize the impact of the flow vectors associated with points with high probability of being objects. On the other hand, if v_i is not associated with any object its weight is maximum $o_i = 1$ because $s_i = 0$. Hence, we take advantage of the image semantics to reduce the FOE estimate error.

Considering these two types of weights, each optical flow vector v_i has an associated weight given by

$$w_i = m_i \cdot o_i. \tag{3}$$

Computing w_i for each optical flow vector v_i, we can estimate the FOE in a non-static scenario. In light of the FOE definition, this is equivalent to finding the closest point to a set of N lines (extensions of the optical flow vectors). Although this problem can be solved via Least-Squares, we estimate the solution point using the Huber Loss [29], as it deemphasizes outliers. Let us define $f(x, L_i)$ as the distance between a point $x \in \mathbf{R}^2$ and a line L_i, parameterized by $L_i = \{p_i + tu_i : t \in \mathbf{R}\}, p_i, u_i \in \mathbf{R}^2, u_i = \frac{v_i}{\|v_i\|}$ for $i = 1, ..., N$, as $f(x, L_i) = \sqrt{(x - p_i)^T (I - u_i u_i^T)(x - p_i)}$. We formulate and solve the following optimization problem

$$\tilde{x} = \underset{x}{\text{argmin}} \sum_{i=1}^{N} \mathcal{L}_\delta \left(\frac{1}{w_i} \cdot f(x, L_i) \right), \tag{4}$$

where $\mathcal{L}_\delta(a)$ is the Huber Loss given by

$$\mathcal{L}_\delta(a) = \begin{cases} \frac{1}{2}a^2, & |a| \leq \delta \\ \delta(|a| - \frac{1}{2}\delta), & otherwise. \end{cases} \tag{5}$$

Solving (4) (with $\delta = 1$) we find the point that minimizes the sum of weighted distances $\frac{1}{w_i} \cdot f(x, L_i)$ for all the obtained optical flow vectors, penalized by the Huber Loss.

Similarly to the previous static case, we perform an iterative refinement of the weighted lines L_i that are considered in this computation. Specifically, we solve problem (4) and then remove the optical flow vectors whose orientation is not according to the optimal FOE found. We repeat this process, solving (4) for a new weight assignment, until it converges, i. e., the difference between the FOE estimates in two consecutive iterations is smaller than a predefined threshold or a maximum number of iterations is achieved.

Exploring the smoothness of the cyclist's trajectory, we perform a weighted average with the FOE of the current and M previous frames as

$$x_t = \frac{\sum_{j=t-M}^{t} \tilde{x}_j \cdot e^{-\tau(t-j)}}{\sum_{j=t-M}^{t} e^{-\tau(t-j)}}, \tag{6}$$

where x_t is the FOE estimate at instant t, $\tilde{x}$ is the minimizer of (4) at the time instant j and τ the decay rate of the weights.

Figure 3 illustrates the intermediate estimates (until convergence) and final FOE (shown in red).

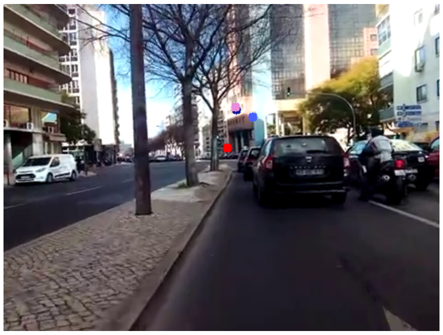

Fig. 3: Intermediate and final (in red) FOE estimates. Light blue shows the estimate given by the Huber Loss without weights, dark blue the Huber Loss estimate considering weights w_i, and pink the estimate after the iterative refinement.

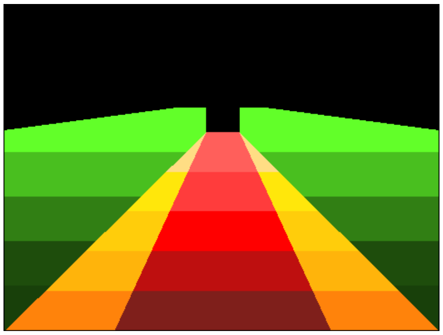

Fig. 4: Division of the 5 different risk zones into 25 sub-regions to promote proximity encoding.

b) Computing the risk descriptor: The obtained FOE gives an estimate of the direction of the cyclist's movement. Based on this direction we can divide the image into five main regions according to the proximity to the cyclist's trajectory. Figures 1 and 4 show these regions, with a color code (red representing the region including the cyclist's predicted trajectory, yellow the region closest to the trajectory, and green the region farther away from this trajectory). The only assumption we make when dividing the image into these regions is that the camera is placed not too far from the ground level, approximately perpendicular to the motion direction, and is not facing up (to the sky).

In order to have a descriptor more spatially fine-grained, we subdivide horizontally each of the previous regions in 5 sub-regions, yielding a total of 25 sub-regions (see Figure 4) as the following expression of the descriptor at instant t shows

$$d_t = \begin{bmatrix} d_t^1 & d_t^2 & \cdots & d_t^{24} & d_t^{25} \end{bmatrix}. \tag{7}$$

The division into these sub-regions encodes the proximity to the cyclist depending on his motion.

Given these motion and proximity aware sub-regions as well as scene object classification and location, we provide a framework to assess dangerous events that can use descriptors based on several criteria: lane occupation, proximity, type of passing by vehicles or combinations of these.

We compute the risk score of each sub-region k at instant

[1] Note that when an object is detected at point p_i, the score s_i for that point coincides with the score s_l for the detected object.

t as:

$$d_t^k = \sum_{l=1}^{N_t^k} r_t^{k,l} \tag{8}$$

where N_t^k is the number of objects in sub-region k at instant t. The risk associated to object l in sub-region k and instant t is given by

$$r_t^{k,l} = \alpha_l \cdot s_l \cdot \gamma_k \cdot \frac{a_t^{k,l}}{b_t^k} \tag{9}$$

where α_l is the object coefficient depending of its type (person, bicycle, car, etc.), s_l the confidence score output by the neural network, γ_k the coefficient of region and sub-region k, $a_t^{k,l}$ the area of object l in sub-region k and b_t^k the area of sub-region k, both at instant t. Note that γ_k codifies the 25 sub-regions and takes into account the larger five regions depicted in Figure 4. Expression (9) combines the fact that different objects (weighted by the classification confidence score output by the neural network) pose different risk levels, and that risk depends on both the cyclist's trajectory and object proximity (given by the regions and sub-regions). Also, the ratio between the area occupied by each object and the total sub-region area informs on how close and how large each object is.

c) Computing distance metric for the descriptor: At this point, our risk assessment framework outputs a risk score for each image sub-region. To provide a more informative assessment and easier to understand by the user, we propose to encode these risk scores in a single global risk level.

We formulate this as a supervised classification problem, and use the Earth Mover's Distance (EMD) metric to perform image retrieval and classify new images in each class [30]. EMD is known to match well perceptual similarities for image retrieval when compared to other distances [30]. If we have intrinsic relations between distribution bins, EMD is a measure of the distance between two distributions and finding the minimum cost that has to be paid to transform one distribution into the other can be cast as a transportation problem. Such formulation is a linear optimization problem for which efficient algorithms are available [30].

In our case, to compare risk events, we wish that sub-regions that are close in the image and belong to the same risk level have small distance, that the distance between two sub-regions increases with the image distance between them and with risk level dissimilarity. Also, we wish to have a small distance between sub-regions that are symmetric in the image. Then, we design a 25×25 distance matrix which assigns distance values between all pairs of sub-regions.

Different descriptors (based on different criteria) can be specified by defining a scale of global risk levels and designing a ground distance matrix (which is an input of the EMD image retrieval) that better models the relation of the criteria and the image locations (sub-regions). In our experimental results (see Section VI) we instantiate this framework, assessing risk based on two separate criteria: lane occupation and proximity.

V. Behavior analysis

Given our intent of creating mobility profiles for the users, we seek to automatically classify the type of transportation taken in each part of a route, sparing user input. For that, we use a supervised learning approach, which relies on labeled data provided by the Bike Monitor app (collected from real users, under real-world circumstances without researcher supervision). Specifically, we use Support Vector Machines (SVMs) as they are a widely used method, very flexible, fast and efficient, do not have many parameters to tune [29].

In this section we describe all the steps of our human activity classification "pipeline", from data acquisition and preprocessing to feature selection. Specifically, after collecting the dataset we preprocess the signals (cleaning and windowing) before extracting relevant features. Once the features are extracted we can perform classification. To maximize accuracy, we also add temporal continuity to the classification [31], due to the continuous nature of the activities in study.

In order to keep a low computational cost without compromising accuracy, we adopt the following strategies: extract the majority of features from time domain signals, choose SVM classifier (known to be computationally efficient), and implement a feature selection method [32] that can drastically reduce the number of features used by the SVM.

We detail the steps of our classification approach below.

d) Data acquisition and preprocessing: We use the Bike Monitor App to collect the signals from the smartphone's sensors (see Section III and [10]). We selected the following signals: linear acceleration along the three axes (X, Y, Z), gyroscope data to compute rotations also along (X, Y, Z), and GPS data to obtain speed. We discard the first and last 10 seconds of each signal to avoid mislabeling, as during these periods the user may be still setting up for the activity or may be already stopped [33].

When selecting a time window it is fundamental that it is long enough to contain the whole activity under analysis, and, on the other hand, short enough that it does not include additional events. Previous works on activity recognition report good accuracy results with sliding windows covering approximately 5 to 10 seconds of movement. Considering our scenario, the app sampling frequency and implementation constraints, features are computed on sliding windows of 100 samples (with a 50% overlap, as this overlap percentage has been successful in the past [33]).

e) Feature Extraction: Feature extraction is a critical step in the design of any classifier. We explore the following statistical features previously used in the literature [33], [31]: mean, standard deviation, root-mean square and mean absolute deviation. In addition to the latter time domain analysis, we extract some frequency domain features using the Fast Fourier Transform, computing the power spectral entropy and spectral energy for each window.

Furthermore, [33] reports that features measuring correlation of acceleration between axes can improve recognition of activities involving multiple body parts. Thus, we also include features encoding the correlation between all pairs of axes.

Finally, we obtain, for each window, a feature vector with a total of 54 features (including time and frequency domain features computed from the 3-axes acceleration, gyroscope and GPS speed signals and the time domain features of the acceleration cross correlation between pairs of axes). Grouping all feature vectors results in the predictor data matrix $\mathbf{X}$.

f) Classification Method - SVM: We use SVMs to classify human activity, based on the previous features, into three classes: cycling, walking and riding a motorized transport (e.g. a car or a bus).

As maximal margin classifiers, SVMs are widely used, benefit from computational advantages over probabilistic methods and are known to perform well on high dimensional data [34]. Although originally designed for binary classification, the One-Versus-All (OVA) and One-Versus-One (OVO) are possible approaches to extend SVMs to multi-class problems [34].

Kernels allow to extend SVMs to cases where the datasets are not linearly separable. This is achieved by kernel functions which translate the original data to a new space, using basis expansions such as polynomials or splines [29].

g) Adding Temporal Continuity and Feature Selection: Although SVMs are effective in classifying individual frames, they do not account for temporal continuity [31]. To this end, we add the generic framework proposed in [31] for incorporating temporal continuity for classification of continuous human activity on top of our SVM classifier. The underlying idea is that probability values computed for a frame at time instant i (f_i) can benefit the classification of successive temporally close frames. Specifically, the probability of a frame f_t belonging to class c is weighted on the temporal distance and similarity between current and past frames. This induces more recent frames to have more impact in the current frame than older ones and assumes that if adjacent frames are identical, then they should belong to the same class (see [31] for details). We add this temporal continuity to the whole set of signals in our dataset.

To keep the classification cost low and prevent overfitting it is important to select relevant features. In this vein, we apply the technique introduced in [32] for feature pruning specifically for SVM, based on Recursive Feature Elimination (RFE). In a nutshell, RFE iteratively trains the classifier and computes a ranking criterion for all features, removing the feature with lowest ranking. Applying RFE to our SVM classifier we are able to significantly reduce the dimensionality of our problem (as we will see in Section VI).

VI. Experimental Results

A. Image-based risk assessment

We tested our risk classification approach for a total of approximately 300 labeled image frames (with close to 100 frames belonging to each of the three risk levels), acquired by the Bike Monitor app by real users. We split our image dataset into training and test set, according to a 75% to 25% ratio.

As we claimed earlier, our risk assessment framework is general and can be applied to different criteria. Here we show results for classifiers based on lane occupation and proximity.

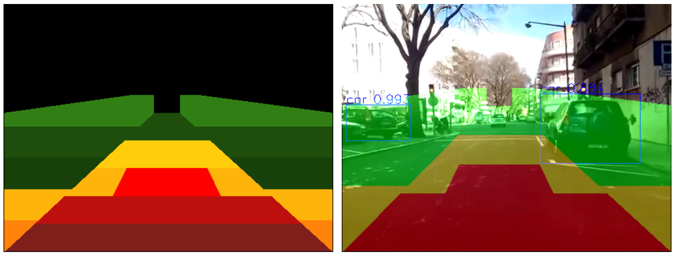

Fig. 5: Proximity risk classifier. a) Proximity regions encoding: red corresponds to the highest risk, yellow to intermediate and green to the lowest risk; b) Proximity based risk regions superimposed on a RGB image.

In the former we study the risk associated with the path occupation or trajectory of the user and define the regions as in Figures 1 and 4, whereas in the latter we assess the risk associated with the proximity of objects to the cyclist, and define the risk regions as shown in Figure 5.

We manually labeled each image according to the levels defined for the two different criteria. In order to assess cyclists' risk based on different criteria one must define risk levels according to the specific criterion and design a distance matrix (used by the EMD) that properly captures the intended notion of nearness.

We define three risk levels (higher level means higher risk). For both classifiers, we consider the risk levels as: 3- the red region is occupied; 2- the yellow region is occupied, and; 1- only the green region is occupied. The distance matrices used for the two classifiers follow the desired properties outlined above. Notwithstanding, we add some alterations to better fit each criterion. The distance matrix used together with the lane occupation criterion adds a multiplicative factor (> 1) to distances between sub-regions belonging to different risk regions (represented by red, yellow or green). On the other hand, for the proximity criterion we add a multiplicative factor (also > 1) when the sub-regions belong to different semi-circular zones, centered on the cyclist.

To estimate the optical flow vectors we used the Lucas and Kanade algorithm [27] with squared windows of 35 pixels, 1 pyramid level, and a 5 skip frame. These parameters depend on the resolution and frame rate; in our experiments we used a resolution of 480×360 and 30fps.

To compute the risk score for each object (see equation (9)) we consider that motorized objects (cars, buses and motorcycles) present higher risk than bicycles which in turn present higher risk than persons. Hence, we assign a higher object type value (1) for motorized vehicles, an intermediate value (0.8) for bicycles, and a lower value (0.6) for person detections. The region risk is defined according to the region to each sub-region belongs to: being higher for sub-regions belonging to the red region, intermediate for sub-regions in the yellow region and lower for sub-regions within the green region. The sub-region risk decreases as we move up vertically in the image (sub-regions near the bottom present higher risk than sub-regions near the top). As object area, instead of directly using the bounding box provided by the Faster-

TABLE I: Risk Classification

(a) Lane occupation based

		Predicted Class		
		1	2	3
Class	1	80	20	0
	2	9.1	81.8	9.1
	3	0	25	75

(b) Proximity based

		Predicted Class		
		1	2	3
Class	1	66.7	33.3	0
	2	10.7	82.1	7.2
	3	0	41.2	58.8

RCNN detection, we computed the area ratio considering the width given by the bounding box, and the height (in pixels) as $max\{0.2 \times \text{bounding box height}, 10\}$. This area is a better approximation of the object projection in the defined risk zones (which map regions on the ground), as we consider that all discoverable objects classes are in contact with the ground. All previous variables belong to the interval $[0, 1]$, yielding a $\text{risk}_{\text{score}}(j)$ for each object also between 0 and 1.

Moreover, we use the Python Toolbox *sklearn* [35] to solve the optimization problem in (4) to estimate the FOE, and the EMD implementation from the *pyemd* [36].

We show the results of our risk classification as a confusion matrix in Table Ia for the Lane Occupation Risk classifier, and in Table Ib for the Proximity Risk classifier. We note that there is no misclassification between risk levels 1 and 3 in any of the classifiers. Thus both classifiers separate well these two extreme classes. The achieved accuracy for the Lane Occupation classifier is relatively high, showing an error rate of 20-25% for each class. For the Proximity classifier, results show some missclassification between risk levels 3 and 2, which we deem to be a result of objects that appear close to the limits of both red and yellow zones upon labeling, and thus incurring some error in the classification. Furthermore, as our risk levels are not continuous, i.e., we have discretized the risk levels throughout the defined areas and not used a smooth continuous risk function, it is expected that the risk classification incurs in some errors when objects are positioned close to the boundaries of each zone.

B. Behavior Analysis

We divided our dataset for behavior analysis (with a duration of approximately 8 hours) keeping again a ratio of approximately 75% of training to 25% of test data [29].

Accuracy is evaluated based on a loss function measuring the classification error for the SVM model, computed using the test examples and the corresponding true class labels. The used loss function is given by $L = \sum_{i=1}^{N} a_i I\{\hat{y_i} \neq y_i\}$, where a_i is the weight of observation i (these weights sum to the respective class prior probability, which are normalized so that all priors sum to one), $I(x)$ is the indicator function, $\hat{y_i}$ is the class label given by the SVM as the class with the maximal posterior probability, and y_i the true class label.

Table II shows the average loss obtained for different kernel functions and parameters C (penalization imposed to points violating the SVM margin).

The SVM classifier achieves highest accuracy (approximately 99%) for $C = 1$ and a linear kernel.

To maximize accuracy, we incorporate temporal continuity by feeding the score of the SVM as input to the method of [31].

TABLE II: Classification error loss for OVA SVM-classification

		C			
		0.5	1	10	20
Kernels	Linear	0.0118	0.0091	0.0754	0.0783
	Gaussian	0.6011	0.5951	0.5951	0.5951
	Polyn. Order 2	0.0236	0.0236	0.0236	0.0236
	Polyn. Order 3	0.0266	0.0266	0.0266	0.0266

TABLE III: Classification error loss when adding temporal continuity

	No temporal cont.	Adding temporal cont.
Linear	0.0091	0.0091
Polyn. Order 2	0.0236	0.0168

We added temporal continuity to the cases that attained higher classification accuracy for the previous “temporal insensitive” SVMs. Table III presents the results obtained.

Adding temporal continuity maintains the highest attained classification accuracy (for the linear kernel), but it increases (by an order of 30%) the accuracy for the Polynomial kernel.

The previous accuracy results were obtained considering the full set of 54 features. Yet, we can reduce the problem dimensionality by applying the feature selection technique SVM-RFE from [32]. Although this method was proposed for the binary case only, we start by selecting features in the $K = 3$ binary classifiers and then we experiment the multi-class case with the most relevant (highest ranked) features found before. With this list of ranked features one can study the impact on the achieved accuracy of eliminating less relevant features, as well as understand what are the most discriminative features (by trying to grasp some intuitive physical interpretation).

Training and testing the previous SVM that maximized accuracy for OVA (with $C = 1$ and linear kernel) and inputting only the 8 most relevant features found (by performing a kind of consensus between the binary classifiers) returns a loss of 0.0647. Hence, we move from $\approx$ 99% accuracy when including all 54 features to $\approx$ 94% after reducing the dimension of the data to 8. This result is very promising, since we can significantly lower the computational load at the cost of a slight accuracy reduction.

We observed that when classifying between walking and another class the losses were very low, suggesting the walking class has very distinctive features with respect to the others. Contrastingly, a significant accuracy degradation was observed when pruning features for the Bike*Vs*Car classifier.

Reducing even more the data dimensionality allows us to visually grasp the multi-class problem. Figure 6 shows all training data reducing the predictor $\mathbf{X}$ to the root-mean square of the speed and the mean rotation along Y (the two most relevant features found by agreement of the 3 binary classifiers). We observe that speed (feature 1) is very effective to separate the classes (it shows remarkably low inner-class variance for Walking). However, we also note that some Bike and Car observations are mixed (these two can originate similar speeds specially within the context of traffic jams).

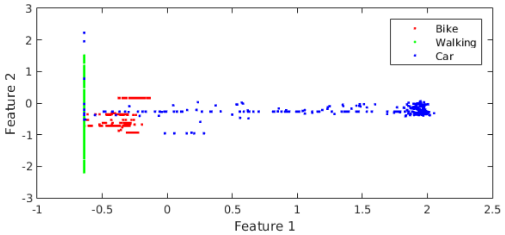

Fig. 6: Scatter diagram of the data reduced to 2 dimensions

VII. Conclusions

In this work we contribute with a novel and complete framework to assess the risk of cyclists' routes. First, we performed improvements on an existing platform (a mobile app that recorded multiple smartphone sensor data), so that we could base our method entirely on videos acquired from the cyclists' smartphones. Then, we proposed a generic framework for image risk descriptors, based on optical flow (to compute the FOE) and semantics, thus being context-aware and invariant to the user. Additionally, we performed behavioral analysis based on smartphone sensor data to automatically detect when the user is riding a bicycle, as opposed to riding a motorized vehicle or walking. Combining several methods from computer vision, image processing, and statistical learning, we were able to overcome many technical hurdles that are common when acquiring and dealing with cyclist's data. Instantiating this framework for two specific criteria (lane occupation and proximity), our risk assessment was shown to perform well for both cases on real data. Similarly, the behavior analysis was also tested on real data and achieved very good accuracy results. Finally, considering this work's potential for city planning and road accidents prevention, we hope it can reach and influence our cities' decision makers.

We identify several possible directions for future work, including taking advantage of the good properties of deep neural networks in our classification problem and making our FOE estimates even more robust. Particularly, we envisage to train our own deep neural network only with images taken from a cyclist's point of view. This way we expect detection and classification of objects to have even higher accuracy. Also, we could benefit from deep learning to classify risk given the FOE and detected objects as input, bypassing the formulation as an image retrieval problem and using EMD. To improve robustness of the proposed FOE estimation method, we could use the dominant direction of each route to help us better prune the optical flow vectors used (as they are the single source of error).

References

[1] "Traffic safety basic facts - main figures," European Road Safety Observatory,, 2016.

[2] J. Strauss, S. Zangenehpour *et al.*, "Cyclist deceleration rate as surrogate safety measure in Montreal using smartphone GPS data," *Accid Anal Prev*, vol. 99, Part A, pp. 287–296, 2017.

[3] S. Gossling, "Urban transport transitions: Copenhagen, city of cyclists." *J. Transp. Geogr.*, vol. 33, no. Dec., pp. 196–206, 2013.

[4] J. Pucher and R. Buehler, "Making Cycling Irresistible: Lessons from The Netherlands, Denmark and Germany," *Transport Reviews*, vol. 28, no. 4, pp. 495–528, 2008.

[5] "Road safety in the european union - trends statistics and main challenges," European Comission,, 2015.

[6] E. Vlahogianni and E. Barmpounakis, "Driving analytics using smartphones: Algorithms, comparisons and challenges," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 196–206, 2017.

[7] I. Cara and E. Gelder, "Classification for safety-critical car-cyclist scenarios using machine learning," in *IEEE ITSC*, 2015, pp. 1995–2000.

[8] A. Muralidharan, C. Flores, and P. Varaiya, "High-resolution sensing of urban traffic," in *IEEE ITSC*, 2014, pp. 780–785.

[9] S. Hasshu, F. Chiclana *et al.*, "Encouraging active commuting through monitoring and analysis of commuter travel method habits," in *SAI Intelligent Systems Conference*, 2015, pp. 179–186.

[10] P. Vieira, J. P. Costeira *et al.*, "Smartcycling: Assessing cyclists' driving experience," in *IEEE I.V.*, 2016, pp. 1321–1326.

[11] L. González, R. Moreno *et al.*, "Learning roadway surface disruption patterns using the bag of words representation," *IEEE Trans. Intell. Transp. Syst.*, vol. PP, no. 99, pp. 1–13, 2017.

[12] F. Seraj, K. Zhang *et al.*, "A smartphone based method to enhance road pavement anomaly detection by analyzing the driver behavior," in *ACM Pervasive and Ubiquitous Computing*, 2015, pp. 1169–1177.

[13] S. Panichpapiboon and P. Leakkaw, "Traffic sensing through accelerometers," *IEEE Trans. Veh. Technol*, vol. 65, no. 5, pp. 3559–3567, 2016.

[14] D. Johnson and M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *IEEE ITSC*, 2011, pp. 1609–1615.

[15] R. Araujo, A. Igreja *et al.*, "Driving coach: A smartphone application to evaluate driving efficient patterns," in *IEEE I.V.*, 2012, pp. 1005–1010.

[16] H. Eren, S. Makinist *et al.*, "Estimating driving behavior by a smartphone," in *IEEE I.V.*, 2012, pp. 234–239.

[17] M. Dozza and A. Fernandez, "Understanding bicycle dynamics and cyclist behavior from naturalistic field data (november 2012)," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 376–384, 2014.

[18] J. Strauss, L. Miranda-Moreno, and P. Morency, "Mapping cyclist activity and injury risk in a network combining smartphone GPS data and bicycle counts," *Accid Anal Prev*, vol. 83, pp. 132–142, 2015.

[19] P. L. Jacobsen, "Safety in numbers: more walkers and bicyclists, safer walking and bicycling." *Inj. Prev.*, vol. 9, no. 3, pp. 205–209, 2003.

[20] H. Aly, A. Basalamah, and M. Youssef, "Automatic rich map semantics identification through smartphone-based crowd-sensing," *IEEE Trans. on Mobile Comput.*, vol. PP, no. 99, pp. 1–1, 2016.

[21] J. Heikkilä and O. Silvén, "A real-time system for monitoring of cyclists and pedestrians," *Image Vis Comput*, vol. 22, no. 7, pp. 563–570, 2004.

[22] H. Zaki, T. Sayed, and A. Cheung, "Computer Vision Techniques for the Automated Collection of Cyclist Data," *Transp Res Rec: Journal of the Transportation Research Board*, vol. 2387, pp. 10–19, 2013.

[23] S. Ren, K. He *et al.*, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[24] S. Pizer, E. Amburn *et al.*, "Adaptive histogram equalization and its variations," *Comput. Vision Graph. Image Process.*, vol. 39, no. 3, pp. 355–368, Sep. 1987.

[25] J. Shi and C. Tomasi, "Good features to track," in *IEEE CVPR*, 1994.

[26] M. Grundmann, V. Kwatra *et al.*, "Calibration-free rolling shutter removal," in *Inter. Conf. on Computational Photography*, 2012.

[27] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI - Vol. 2*, 1981, pp. 674–679.

[28] J. Lee and A. Bovik, "Estimation and analysis of urban traffic flow," in *IEEE ICIP*, 2009, pp. 1157–1160.

[29] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, ser. Springer series in statistics. New York: Springer, 2009.

[30] Y. Rubner, C. Tomasi, and L. J. Guibas, "Earth mover's distance as a metric for image retrieval," *IJCV*, vol. 40, no. 2, pp. 99–121, 2000.

[31] N. Krishnan and S. Panchanathan, "Analysis of low resolution accelerometer data for continuous human activity recognition," in *IEEE ICASSP*, 2008, pp. 3337–3340.

[32] I. Guyon, J. Weston *et al.*, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[33] L. Bao and S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Pervasive Computing*, pp. 1–17, 2004.

[34] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[35] F. Pedregosa, G. Varoquaux *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[36] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *IEEE ICCV*, 2009.