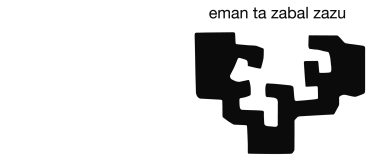


Máster Universitario en Sistemas Inteligentes e Ingeniería Computacional

Master Thesis

Driver Drowsiness Detection in Facial Images

Jorge Reta Cárcamo

Director:
Fadi Dornaika

Co-director:
Ignacio Arganda Carreras

2018

Abstract

Driver fatigue is a significant factor in a large number of vehicle accidents. Thus, drowsy driver alert systems are meant to reduce the main cause of traffic accidents. Different approaches have been developed to tackle with the fatigue detection problem. Though most reliable techniques to asses fatigue involve the use of physical sensors to monitor drivers, they can be too intrusive and are less likely to be adopted by the car industry. A relatively new and effective trend consists on facial image analysis from video cameras that monitor drivers.

How to extract effective features of fatigue from images is important for many image processing applications. This project proposes a face descriptor that can be used to detect driver fatigue in static frames. This descriptor represents each frame of a sequence as a pyramid of scaled images that are divided into non-overlapping blocks of equal size. The pyramid of images is combined with three different image descriptors. The final descriptors are filtered out using feature selection and a Support Vector Machine is used to predict the drowsiness state. The proposed method is tested on the public NTHUDDD dataset, which is the state-of-the-art dataset on driver drowsiness detection.

Acknowledgements

I would like to thank my directors, Dr. Fadi Dornaika and Dr. Ignacio Arganda Carreras, for their support, guidance with their expert knowledge and for their patience with all the hindrances faced during this thesis.

I specially wish to thank Antía because without her support and constant motivation this project would not have been possible.

Table of contents

Index of figures

Index of tables

1.

Introduction

Car accidents are one of the biggest concerns in society as they lead into thousands of deaths all over the globe. Drowsiness and fatigue are one of the main causes of distraction in drivers, a situation that dramatically increases the probability of suffering a car accident.

Fatigue can be detected if some of the following symptoms are observed: yawning, difficulty keeping eyes open, head or body nodding, feeling depressed and irritable, difficulty maintaining concentration, slower reaction and responses, vehicle wandering from the road or into another lane [5]. For each driver, different symptoms and with different degrees may be detected. As an attempt to quantify those symptoms, the Karolinska Sleepiness Scale (KSS) [6] is a method to score drivers' drowsiness level. This scale is based on subjective self perception of the alertness state and consists on nine levels that cover from an *Extremely alert* state to a *Very sleep or great effort to keep alert* state.

Therefore, unlike infractions such as driving under the effects of alcohol, detecting drowsiness in drivers is a harder problem that requires to develop complex solutions. Most of them rely on hand-crafted features that can be extracted from the driver of from the vehicle itself.

1.1. Objectives

The main objective of this project is the development of a facial image descriptor and classification algorithm capable of detecting drowsiness of drivers in different ambient

conditions.

This objective is achieved by a feature extraction stage that comprises three different image descriptors: Covariance descriptor, Histogram of Oriented Gradients and Local Binary Pattern, in conjunction with a multilevel and multiblock image representation known as Pyramid Multilevel descriptor. A second stage of classification includes a robust classifier such as Support Vector Machines.

1.2. Related work

In order to detect drowsiness, most of the techniques can be grouped into three categories. The first one focuses on monitoring the driver's behavior, like grip force on the steering wheel, speed of the vehicle, acceleration, braking or gear changing [7]. The second category makes use of physiological information such as heart rate, electrocardiogram (ECG), electroencephalogram (EEG) and electrooculogram (EOG) and blood pressure [8, 9]. The third category is based on computer vision using cameras and optical sensors to extract features and then analyze whether the driver is in the state of fatigue. Different approaches have been developed on computer vision, covering mainly facial features such as eye blinking [10, 11, 12], yawning [13] or head nodding [14]. These last approaches have a great dependence on an accurate location of eyes and mouth, a requisite that can be very challenging in a real-life situation. In [3], three existing neural networks are used to extract facial features and combined with a Support Vector Machines classifier to predict drowsiness. In [15] a Multi Granularity Convolutional Neural Network (MCNN) is used to extract facial features and a Long Short Term Memory (LSTM) network is used to classify the drowsiness level.

It is assumed that drowsiness will manifest as rapid and constant blinking, nodding or head swinging, and frequent yawning. One of the simplest method to predict drowsiness level is to set a threshold on extracted drowsiness-related symptoms. PERcent of Eye CLOSure (PERCLOS) is a video-image-based method to track eye closure. It is calculated as the total time that the driver's eyelids are closed [16].

[17] showed that when the head inclination angle exceeds a certain value and duration, the level of alertness of the driver is lowered. In [18], yawning is detected based on the rate of change of the mouth contour and is determined as the only sign of drowsiness. This approach may encounter false-alarms when the required visual cues cannot be distinguished from the similar motions, e.g. talking or laughing. In [19], the authors developed a gaze

zone detection algorithm based on features learnt using a convolutional neural network. Based on these features, support vector machine (SVM) is used to estimate driver gaze zone. In addition to the mentioned works, some researchers consider the texture dynamics [20, 8].

2.

Face Descriptors

In this chapter, a brief description of the used image descriptors are presented. Three handcrafted descriptors —Covariance, Histogram of Oriented Gradients and Local Binary Patterns— are combined with the Pyramid Multi-Level descriptor in order to produce more performing descriptors.

2.1. Pyramid Multi-level Descriptor (PML)

The PML Descriptor (PMLD) representation introduced in [21] adopts an explicit pyramid representation of the original image, representing the image at different scales. For each scale, the image is divided into rectangular blocks. The descriptor of each block is extracted at every level and the final descriptor of the image is the concatenation of the descriptors from each block.

More precisely, let I be a $W \times H$ image and let P be its pyramid representation with ℓ levels, so $P = \{P_1 \dots P_\ell\}$. The level ℓ corresponds to the original image. The size of the image at each level of the pyramid representation will be $w_{i-1} = w_i \times \frac{i-1}{i}$ and $h_{i-1} = h_i \times \frac{i-1}{i}$ with $i = 1 \dots \ell$. At each level P_i, the image is divided into i^2 blocks and the image can be represented as $P_i = \{B_{i,1}, \dots, B_{i,n_i}\}$, with $n_i = i^2$. The size of each block is constant along all pyramid representation, $\frac{W \times H}{\ell^2}$.

The pyramid representation of image I by ℓ levels is the concatenation of ℓ sequences

L_i $(i = 1, \ldots, \ell)$, representing each on of the blocks of every level.

$$L_i = \{B_{i,1}, \ldots, B_{i,n_i}\} \tag{2.1}$$

Let $\phi(B)$ be a descriptor extracted from block B. Then, the descriptor of the image at level i can be expressed as:

$$\phi(L_i) = \phi(B_{i,1})|| \ldots ||\phi(B_{i,n_i}) \tag{2.2}$$

where the || denotes the concatenation.

Therefore, the ℓ-PML descriptor of an image I can be defined as

$$\ell - PMLD = \phi(L_1)|| \ldots ||\phi(L_\ell) \tag{2.3}$$

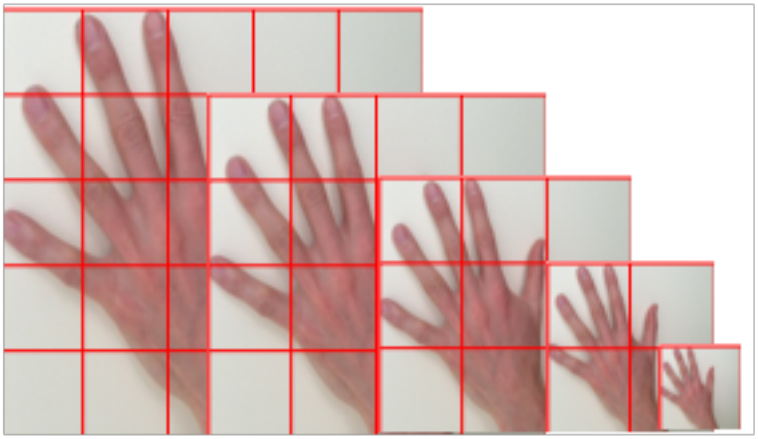


Figure 2.1: Example of PML for 5 levels. The image is scaled in 5 levels and divided in square blocks for each level.

2.2. Covariance descriptor

The covariance descriptor was proposed by [22] for generic object detection and texture classification tasks. Opposed to histogram based descriptors, the covariance descriptor computes covariance matrices of the color channels and the spatial derivatives, resulting in a low-dimensional descriptor.

The algorithm can be easily extended with new features provided they can be presented spatially. In this case, the Local Binary Pattern (LBP) and Local Phase Quantization (LPQ) images are also included in the covariance matrix as well as the aforementioned color channels and spatial derivatives.

Let I be an image and let F be the $W \times H \times d$ dimensional feature extracted from I with any mapping function, $\phi(\cdot)$, like intensity image, RGB color image or LBP image.

$$F(x,y) = \phi(I,x,y)$$

For a given rectangular region, $R \subset F$, let $\{\mathbf{z}_k\}_{k=1\ldots n}$ be the d -dimensional feature points inside region R. It is possible to represent the region R by the $d \times d$ covariance matrix of the feature points.

$$\mathbf{COV}_R = \frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{z}_k - \boldsymbol{\mu})(\mathbf{z}_k - \boldsymbol{\mu})^T \tag{2.4}$$

where μ is the mean value of the points.

2.3. Histogram of Oriented Gradients descriptor

Histogram of Oriented Gradients (HOG) is an extension of SIFT (Scale Invariant Feature Transform) presented in [23]. The main idea is that local object appearance and shape can be characterized by the distribution of local intensity gradients or edge detections.

The image is divided into small overlapping spatial regions or *cells*. Then, for each cell the 1-D histogram of gradient orientation is accumulated over all the pixels in the cell. This creates the descriptor of the cell. The HOG descriptor of the image is formed by the combination of the descriptors of the cells.

This method can be explained in five stages:

The first stage applies an optional global image normalization equalization intended to reduce the influence of illumination effects.

The second stage computes first order image gradients. These capture contour, silhouette and some texture information, while providing further resistance to illumination variations

The third stage aims to produce an encoding that is sensitive to local image content while remaining resistant to small changes in pose or appearance. Gradient orientation information is pooled locally in the same way as the SIFT feature.

The fourth stage computes normalization, which takes local groups of cells and contrast normalizes their overall responses before passing to next stage. Normalization introduces better invariance to illumination, shadowing, and edge contrast. It is performed by accumulating a measure of local histogram "energy" over local groups of cells that also called "blocks". The normalized block descriptors are referred as Histogram of Oriented Gradient (HOG) descriptors.

The final step collects the HOG descriptors from all blocks of a dense overlapping or non-overlapping grid of blocks covering the detection window into a combined feature vector.

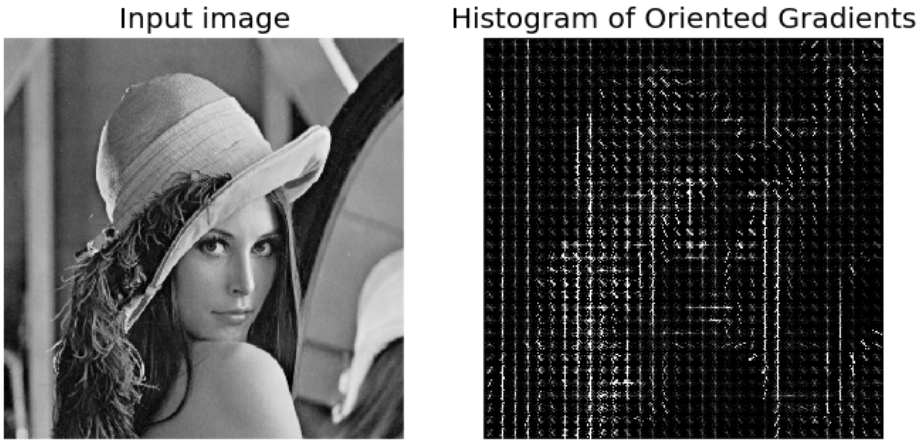


Figure 2.2: Example of HOG descriptor

2.4. Local Binary Pattern descriptor

Local Binary Pattern (LBP) is a texture descriptor presented in [24] that works on gray scale images and allows a very simple and efficient way to encode and image. Due to its reduced computational and its discriminative ability, LBP has been widely used in texture processing and analysis.

LBP works encoding each pixel in an image analyzing its neighborhood.

$$LBP_{P,R}(g_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p,\ s(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases} \tag{2.5}$$

where P is the number of neighbors, R is the size of the neighborhood and g_c and g_p are the grey intensity of central pixel an the p pixels in the neighborhood.

For each neighborhood, the intensity of the central pixel is compared to its neighbors. The final image descriptor is the histogram of LBP codes in the image.

2.4.1. Rotation invariance

An improvement of the classic LBP descriptor is the rotation invariant pattern. Local Binary Pattern as described in section 2.4 is not invariant to rotation. The same neighborhood of a pixel yields a different LBP descriptor if the neighborhood is rotated. The rotation invariant procedure was proposed to compensate the weakness of LBP against rotations of the neighborhood of each pixel. The resulting descriptor is the smallest LBP value of a pixel after applying all possible rotations to its neighborhood.

$$LBP^{ri}_{P,R}(g_c) = min\{ROR(LBP_{P,R}, i) \mid i = 0, \ldots, P-1\} \tag{2.6}$$

where $ROR(x,i)$ is a binary right shift of i pixels in the neighborhood.

2.4.2. Uniform pattern

The same authors of [24] noticed that most part of the relevant information of a texture can be described by the *so*-called uniform patterns. An LBP code is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the

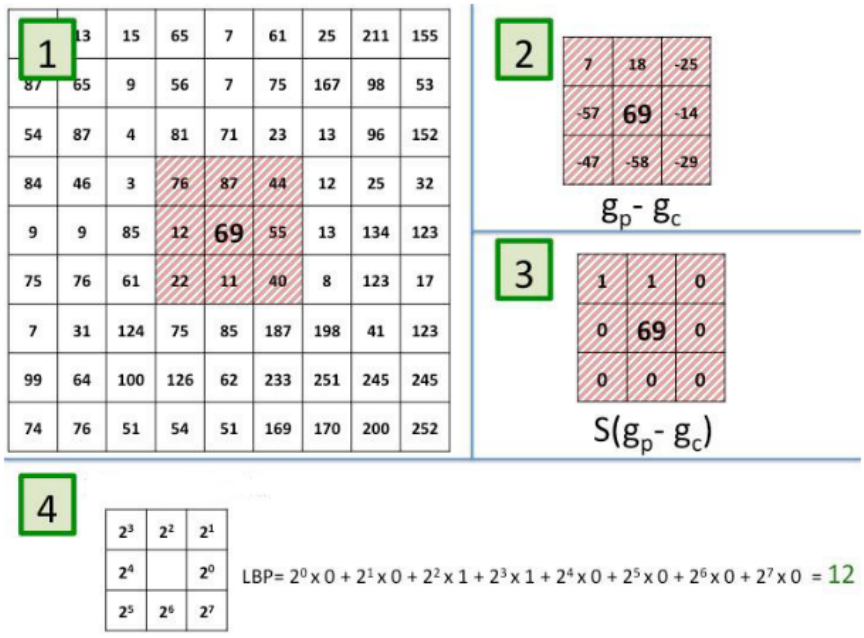


Figure 2.3: Example of LBP for a pixel: 1) example image and selected neighborhood, 2) values of $g_c - g_p$, 3) function $s(x)$ for each pixel in neighborhood, 4) LBP value for selected pixel.

bit pattern is considered circular. In Figure 2.4 an example of uniform and non-uniform patterns is presented along with the number of transitions in each pattern.

Considering only these patterns, any pixel in an image can be described with $P(P-1)+2$ different values.

$$LBP^{riu2}_{P,R}(g_c) = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P+1 & \text{otherwise} \end{cases} \tag{2.7}$$

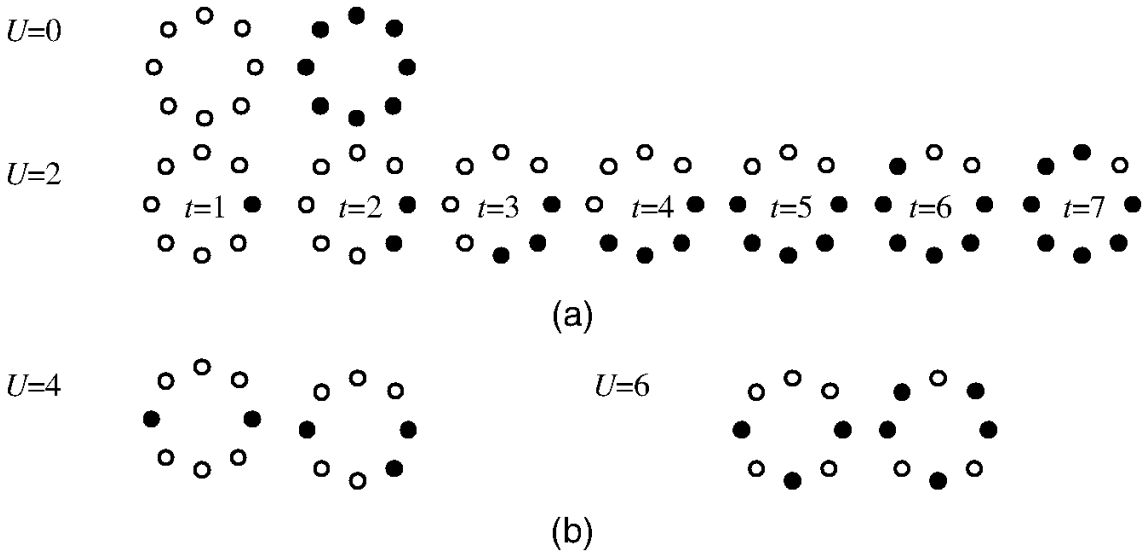


Figure 2.4: Uniform and non-uniform patterns. a) Example uniform patterns with at least $U = 2$ bitwise transitions. b) Example of patterns with more than $U = 2$ bitwise transitions that are not uniform. Source: [1].

3.

Feature Processing

3.1. Principal Component Analysis (PCA)

Principal Component Analysis is an unsupervised feature extraction statistical procedure that searches for k n-dimensional orthogonal vectors that can be best used to represent the original data, where $k \leq n$. The original data are then projected onto a much smaller space, resulting in a dimensionality reduction of data.

Let $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_n}]^\mathbf{T} \in \Re^{n \times m}$ be a collected data matrix, where $\mathbf{X}$ is composed of m-dimensional n data vectors $\mathbf{x_i} \in \Re^\mathbf{m}$. After applying z-score standardization to each dimension, $\mathbf{X}$ can be decomposed using singular value decomposition as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T \tag{3.1}$$

where $\mathbf{T} = [\mathbf{t_1}, \ldots, \mathbf{t_m}] \in \Re^{n \times m}$ and $\mathbf{P} = [\mathbf{p_1}, \ldots, \mathbf{p_m}] \in \Re^{m \times m}$ consist of score vectors $\mathbf{t_j} \in \Re^n$ and orthogonal loading vectors $\mathbf{p_j}$, respectively. The z-score standardization is intended to reduce the influence of different scales of the features, so each feature is transformed to the same range using its mean and standard deviation: $\mathbf{Z} = \frac{\mathbf{X} - \mu}{\sigma}$. The vectors $\mathbf{p_j}$ are eigenvectors of covariance matrix Σ defined as

$$\Sigma = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} = \mathbf{P}\Lambda\mathbf{P}^T \tag{3.2}$$

where $\mathbf{P}\mathbf{P}^T = \mathbf{P}^T\mathbf{P} = \mathbf{I}_m$ and $\Lambda = diag(\lambda_1, \ldots, \lambda_m)$ is a diagonal matrix whose diagonal

components sorted in descending order (i.e., $\lambda_1 > \ldots > \lambda_m$) are eigenvalues of Σ. The matrix Λ and λ_i are defined as

$$\begin{aligned} \Lambda &= \frac{1}{n-1}\mathbf{T}^T\mathbf{T} = diag\{\lambda_1,\ldots,\lambda_m\} \\ \lambda_j &= \frac{1}{n-1}\mathbf{t}_j^T\mathbf{t}_j \;\; (j=1,\ldots,m) \end{aligned} \tag{3.3}$$

In other words, λ_j is the variance of n projections of data vector x_i, $i = 1,\ldots,n$ onto eigenvector p_j. In PCA, dimensionality reduction is performed by selecting l eigenvectors that correspond to the largest l eigenvalues among m eigenvalues sorted in decreasing order.

3.2. Fisher Score

Fisher Score is a supervised feature selection method which uses classes to identify features with best discriminant abilities. The key idea of Fisher Score [25] is to find a subset of features, such that in the data space spanned by the selected features, the distances between data points in different classes are as large as possible, while the distances between data points in the same class are as small as possible.

Let $\mathbf{X}$ be a set of data and $\mathbf{x}_j$ each feature, with $j = 1\ldots m$. The Fisher Score for each feature can be computed as follows:

$$F(\mathbf{x}_j) = \frac{\sum_{k=1}^{c} n_k(\mu_k^j - \boldsymbol{\mu})^2}{(\sigma^j)^2} \tag{3.4}$$

where c is the total number of classes, n_k is the number of instances of class k, μ_k^j σ_k^j are the mean of and standard deviation of k-th class for feature j, $\boldsymbol{\mu}$ is the total mean of the data and $(\sigma^j)^2 = \sum_{k=1}^{c} n_k(\sigma_k^j)^2$. Once the score is computed for each feature, they can be rearranged accordingly in descending order and the top-m ranked features shall be selected.

4.

Classification

A wide variety of classification algorithms has been used in the academic and industry fields, like Convolutional Neural Networks or Support Vector Machines. In this project, Support Vector Machines is selected among other algorithms due to its robustness and to the realtively short time needed for training in compared to algorithms like neural networks.

4.1. Support Vector Machines (SVM)

Support Vector Machines (SVM) [26] is an algorithm used for linear and non linear classification. The SVM algorithm searches boundaries with the maximum margin of separation from the training data mapped into a space. Margin maximization usually reduces the generalization error (i.e., the expected error on a test set independent from the dataset used to build the classifier).

Let D be a set of linearly separable data $\{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$ which contains d-dimensional input vectors $\mathbf{x} \in \Re^d$ and their corresponding classes $y \in \{-1, +1\}$. The hyperplane that separates with the maximum margin of separation can be expressed as $g(\mathbf{x}) = \omega^T \mathbf{x} + b$. This is an optimization problem that can be written in terms of two hyperplanes ($g(\mathbf{x}) = 1$ and $g(\mathbf{x}) = -1$).

$$\begin{aligned} g(x) \geq 1, \text{if } y = +1 \\ g(x) \leq 1, \text{if } y = -1 \end{aligned} \tag{4.1}$$

The margin of separation between the parallel hyperplanes is $\frac{2}{\|\omega\|}$.

Maximizing the margin is equivalent to finding a solution to the following problem

$$\begin{aligned} \min_{\omega,b} \quad & \frac{1}{2}\omega^T\omega \\ s.t. \quad & y_i(\omega^T\mathbf{x_i}+b) \geq 1 \;\; i=1,\ldots,n \end{aligned} \tag{4.2}$$

When data is not linearly separable, this problem can be generalized by means of a slack variable ε and a regularization parameter $\mathcal{C}$, resulting in the following formulation

$$\begin{aligned} \min_{\omega,b} \quad & \frac{1}{2}\omega^T\omega + \mathcal{C}\sum_{i=1}^{n}\varepsilon_i \\ s.t. \quad & y_i(\omega^T\mathbf{x}_i+b) \geq 1-\varepsilon_i, \;\; \varepsilon_i > 0, \;\; i=1,\ldots,n \end{aligned} \tag{4.3}$$

where ε is related to the degree of error allowed and $\mathcal{C}$ controls the importance that is attributed to the second term included in the minimization objective, to allow an error margin.

This problem can be reformulated by the Lagrange formalism [27] to a corresponding dual form as

$$L_p = \frac{1}{2}\omega^T\omega + \mathcal{C}\sum_{i=1}^{n}\varepsilon_i - \sum_{i=1}^{n}\alpha_i(y_i(\omega^T\mathbf{x}_i+b)-1+\varepsilon_i) - \sum_{i=1}^{n}r_i\varepsilon_i \tag{4.4}$$

where $\alpha_i \geq 0$ and $r_i \geq 0$ are Lagrange multipliers; r_i are introduced to ensure positivity of ε_i. Differentiating with respect to ω and b results in the dual form of the Lagrangian problem

$$\begin{aligned} \max_{\alpha_i} \;\; & L_D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j \\ s.t. \;\; & \sum_{i=1}^{n}\alpha_i y_i = 0, \;\; 0 \leq \alpha_i \leq \mathcal{C} \end{aligned} \tag{4.5}$$

All the formulation has been presented with a linear kernel $\mathbf{x}_i\mathbf{x}_j$, although it is possible to introduce other functions instead of the linear kernel that expand the feature space and make it possible a better separation of data. One of the most useful kernels is radial basis kernel (RBF)

$$K(\mathbf{x}_i,\mathbf{x}_j) = e^{(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{\sigma^2})} \tag{4.6}$$

This kernel has one extra parameter that has to be optimized, apart from $\mathcal{C}$, which is σ.

5.

Proposed approach

In this chapter, four different proposed approaches are presented, each of one trying to obtain the best performance of the classification. Performance of each approach is computed over the validation set.

The main structure of each approach is very similar and the main differences between them lie on the type of descriptor used.

5.1. Single descriptor

In Figure 5.1, the pipeline for a single descriptor is shown. The descriptor is a PML image representation in combination with one of the three descriptors described in the Face Descriptors chapter.

The first step consists in the descriptor extraction for each frame in the videos, as a combination of the aforementioned PML descriptor and COV, HOG or LBP descriptor.
The second step is a dimensionality reduction through Principal Component Analysis that removes more than 95 % of the features in the training set.
The third step is intended to reduce even more the input data and select the most relevant features according to Fisher Score.
Finally, the last step trains a SVM classifier.

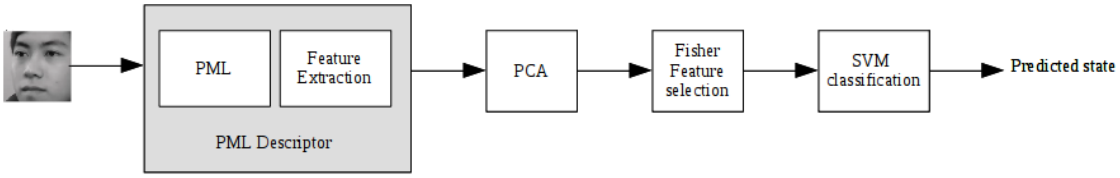


Figure 5.1: Pipeline for a single descriptor.

5.2. SVM blending

Another approach that can benefit of the results obtained form each individual classifier from Section 5.1 is shown in Figure 5.2. Three independent classifiers are built following the procedure for a single descriptor are run in parallel, so each classifier outputs a predicted frame label using SVM. Finally, the predicted output is the average over the three classifiers.

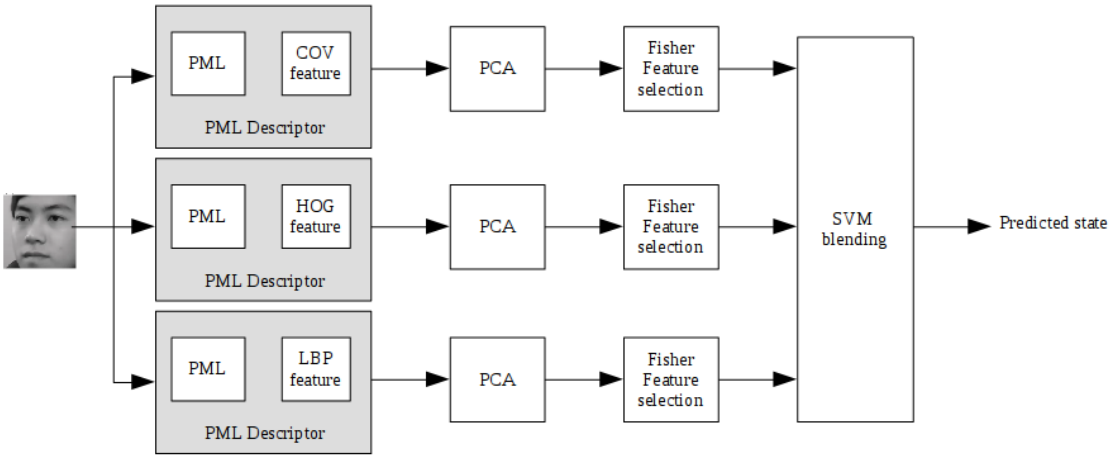


Figure 5.2: Pipeline for blending SVM.

5.3. Concatenation of descriptors

This approach applies the three hand-crafted descriptors for each video frame and builds up a new descriptor as the concatenation of each individual descriptor.
A similar pipeline is applied as in the case of a single descriptor, which comprises PCA dimensionality reduction, feature selection by Fisher Score and SVM classification. On Figure 5.3, the scheme of this approach is shown.

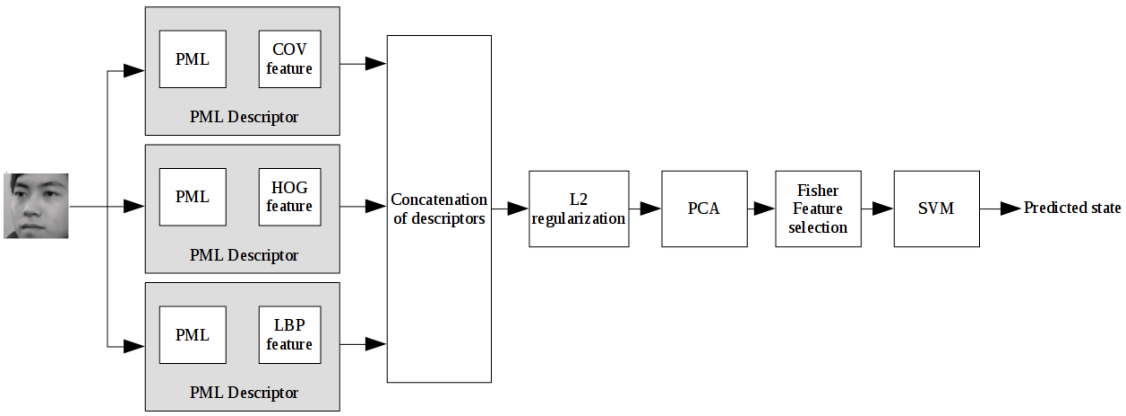


Figure 5.3: Pipeline for combination of descriptors

5.4. Concatenation of reduced descriptors

As a variant of the proposed approach in Section 5.3, this last proposal firstly reduces the individual descriptors by PCA and then concatenates the resulting descriptors to form a new descriptor.

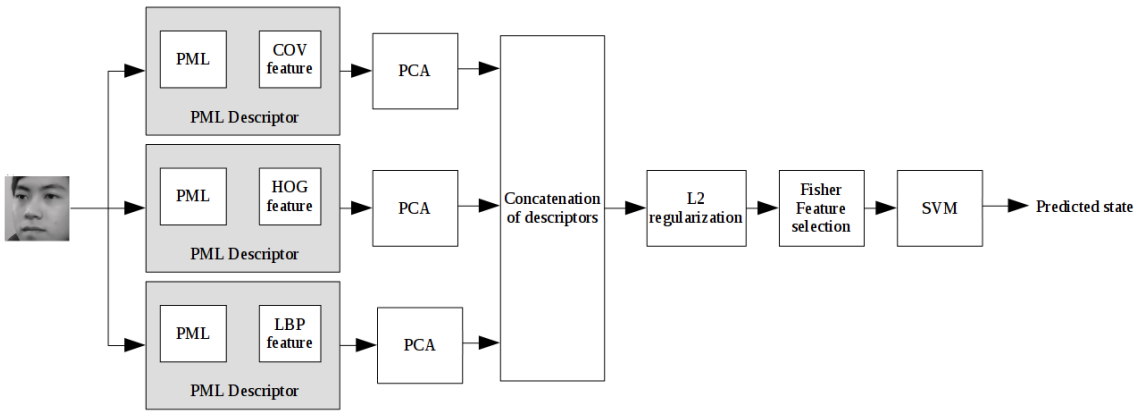


Figure 5.4: Pipeline for combination of descriptors after individual reduction

6.

Experimental setup

In this chapter, the experimental setup is presented, including a detailed description of the dataset, the way that it has been processed, how the dimensionality reduction has been applied and the feature selection process. Also, an explanation of the optimization procedure that has been followed to determine the best classification model and its parameters is included.

6.1. Dataset description

Despite the importance of research in a practical drowsy driver detection system, most research have used relatively limited datasets. The generalization of different approaches to drowsy driver detection analysis remains unknown. In the absence of performance evaluation on a common public dataset, the comparative strength and weakness of different approaches is difficult to determine. Furthermore, most of the proposed approaches have drawbacks due to impractical reasons or do not provide sufficient discrimination to capture the uncertainties. Moreover, most of the existing methods do not evaluate the robustness of their system against subjects from different ethnicities, races, genders, various illumination conditions and partial occlusion (e.g. glasses, sun-glasses and facial hair).

For this thesis, the public dataset NTHU Drowsy Driver Detection (NTHUDDD) has been used [3]. This dataset contains 36 subjects, including different gender and ethnicity, in five different situations: bareface, wearing glasses, night bareface, wearing sunglasses and night wearing glasses, as shown in Figure 6.1. The total dataset is divided into three

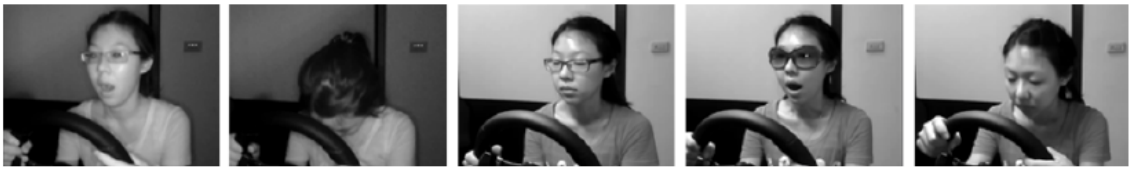


Figure 6.1: Example frames of different situations (nightglasses, night bareface, glasses, sunglasses and bareface).

subsets: training, validation and test. The test set is not yet publicly available, so all the experimentation is done using the training and validation sets.

The training set consists on 18 subjects with a total of 360 video clips (722,223 frames) and the validation set consists on 4 subjects with a total of 20 video clips (173,259 frames). All videos are labelled in two classes, drowsy and non drowsy driver state. The dataset has been previously processed so a in-face cropped video is generated for each original video, resulting in a dataset of 250×250 video pixels. Also, for each frame, a label is provided reporting whether face detection and cropping has been accomplished or not.

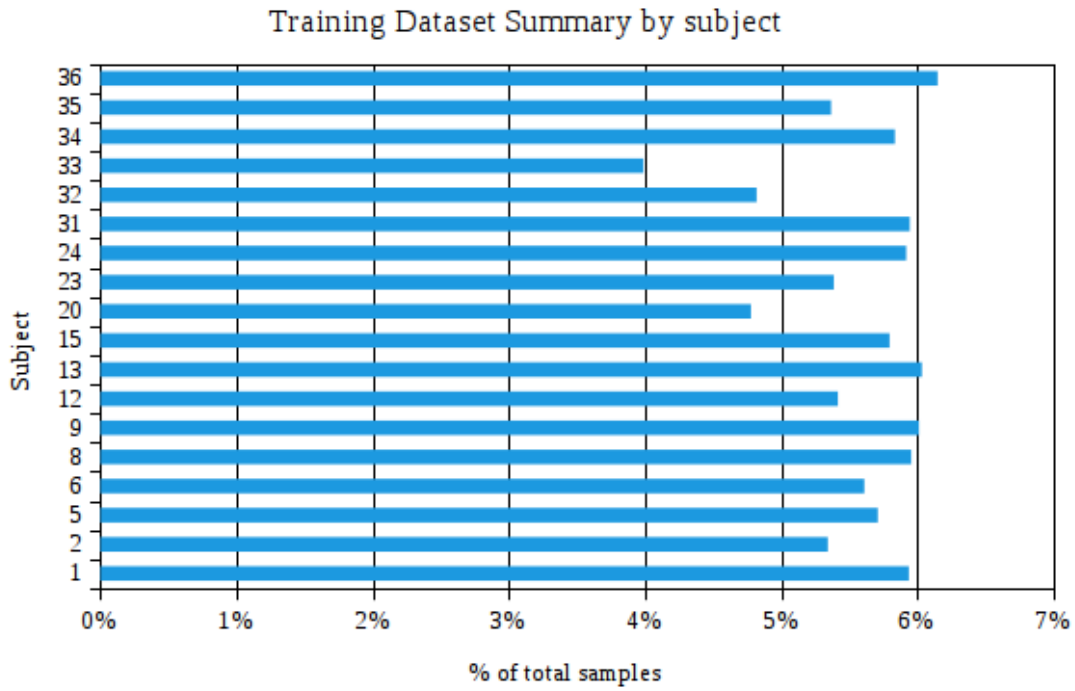
The videos in each situation of the training set are divided into 4 categories according to the subject's behavior: non drowsy state, drowsy state, yawning and head nodding. In this work, the last three states are considered the drowsy class. In Table 6.1 an example of how the videos are distributed per subject can be seen.

subject	bareface	glasses	sunglasses	night bareface	nightglasses	Total
Training subject	4	4	4	4	4	20

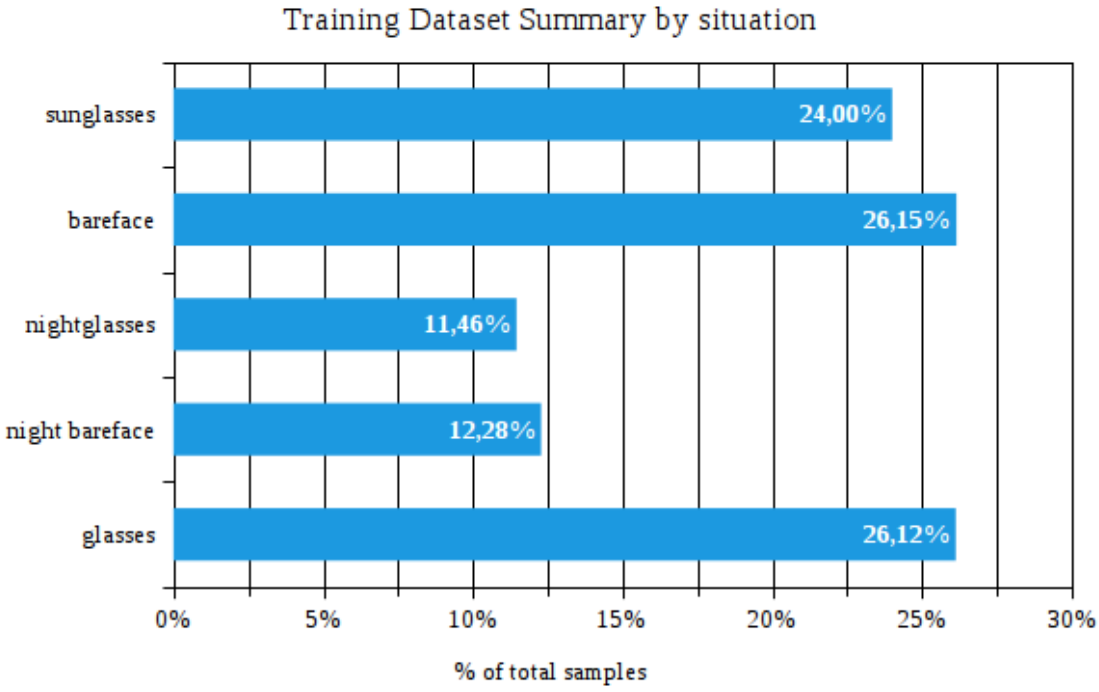
Table 6.1: Example of the distribution of videos per subject in the training set.

A summary of the training set is shown in Figure 6.2. The set is divided in 5 subsets, one for each situation resulting in 5 independent problems. The training set has been subsampled at a $1/10$th rate to reduce the training data size. After the subsampling process, all frames that do not have a matching positive label in the face detected list have been removed. The resulting dataset samples are shown in Table 6.2.

The videos in the validation set are only grouped by subject and situation, so there is no distinction in subject's behavior. Subsampling has not been applied to these videos, but every frame that do not have a matching positive label in the face detected list has been removed. A summary of the number of samples is shown in Table 6.3 and in Figure 6.3.



(a) Distribution of the training samples by subject.



(b) Distribution of the training samples by situation.

Figure 6.2: Summary of training sample distribution.

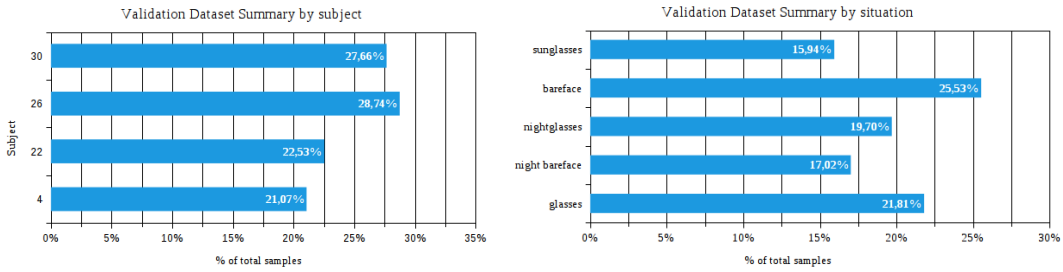
subject	bareface	glasses	sunglasses	night bareface	nightglasses	Total
1	903	960	902	507	498	3770
2	794	955	757	420	466	3392
5	1128	1040	1021	436	0	3625
6	897	931	868	433	433	3562
8	981	923	940	479	458	3781
9	960	996	931	466	464	3817
12	907	892	798	467	375	3439
13	999	996	856	499	481	3831
15	931	986	902	427	434	3680
20	846	798	603	397	390	3034
23	932	882	784	345	477	3420
24	998	1056	874	411	419	3758
31	975	958	904	473	464	3774
32	799	775	754	391	341	3060
33	683	646	651	286	266	2532
34	994	963	861	465	422	3705
35	907	853	818	441	388	3407
36	966	977	1013	451	497	3904
Total	16600	16587	15237	7794	7273	63491

Table 6.2: Training samples per situation.

6.2. Face Alignment

The videos in the NTHUDDD dataset have not been taken from a front perspective and the image in the scene contains more information than the strictly needed like different backgrounds or the steering wheel. Since only the facial image and subject behavior are needed to detect a drowsy state, a preprocessing stage is required on each video so the faces can be aligned and scaled to a fixed size. The dataset used in this work has been provided already face aligned.

The face alignment is done locating the 2D position of the eyes using the Ensemble Regression Tree (ERT) [28]. The coordinates of the eyes are used to compensate the in-plane rotation of the face. Finally, the image is scaled such that the inter-ocular distance is normalized to a fixed value l and the face aligned image is of size 250×250 pixels.



(a) Validation samples by subject.

(b) Validation samples by situation.

Figure 6.3: Distribution of validation samples.

	Subject				
Situation	004	022	026	030	**Total**
bareface	10202	15632	10799	4222	40855
glasses	3043	2519	15518	13819	34899
sunglasses	3504	2399	11661	7940	25504
night bareface	11633	7603	5564	2431	27231
nightglasses	5324	7899	2448	15844	31515
Total	33706	36052	45990	44256	160004

Table 6.3: Validation set samples per subject and situation

6.3. Image descriptors

Each of the image descriptors introduced in Chapter 2 has been applied in combination with the PML descriptor. Selection of the image descriptors parameters has been done based on best experimental results. The number of pyramid levels selected is 5, according to previous work on groups of scaled images for the SIFT descriptor [29].

Figure 6.4 illustrates the procedure that leads to a processed video database for each image descriptor.

6.3.1. COV descriptor

As described in Section 2.2, the covariance descriptor can be extended with new features. In this thesis, the covariance descriptor includes not only color channel and position featu-

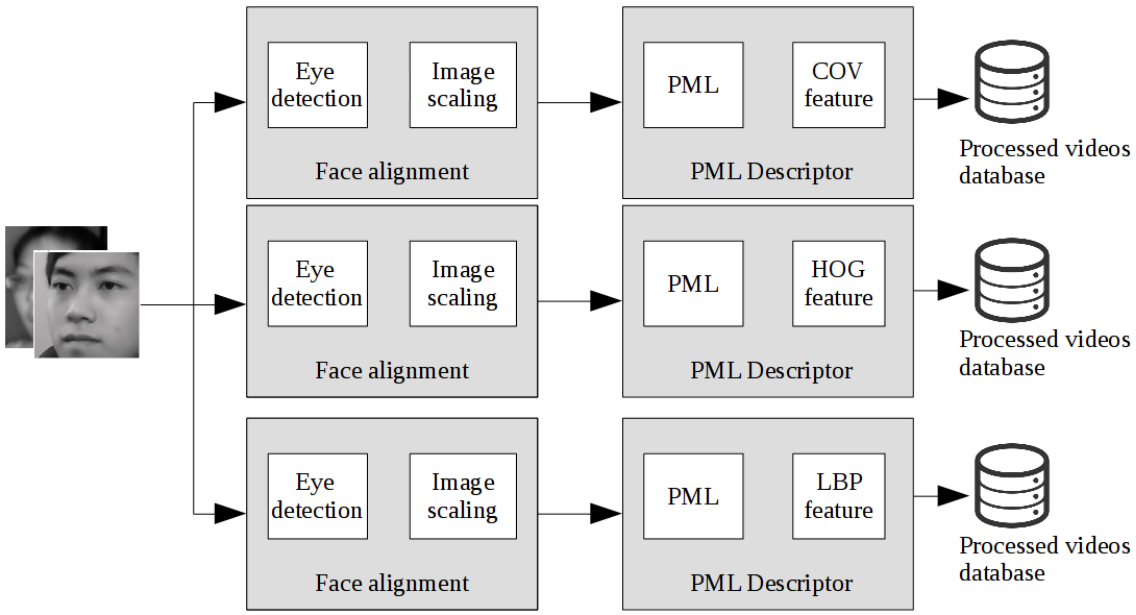


Figure 6.4: Workflow of video database computing. The pipeline operations comprise (i) reading each video file, (ii) converting the video into frames, (iii) processing each frame with every image descriptor and (iv) saving all processed videos in a database.

res, but also local binary pattern descriptors. The list of features included in the covariance descriptor comprises:

- (x, y) coordinates
- Spatial first and second order gradients
- RGB color channels
- HSV color channels
- LBP uniform
- LBP rotation invariant
- LBP uniform rotation invariant

Each image size is 250×250 and the block size is 50×50, resulting in a pyramid structure as shown in Table 6.4.

Pyramid level	Image Size	Descriptor Length
5	250×250	5250
4	200×200	3360
3	150×150	1890
2	100×100	840
1	50×50	210
Total		11550

Table 6.4: Pyramid structure of COV descriptor.

6.3.2. HOG descriptor

In this case, a fixed block size of 32×32 is selected, so the original frames are scaled to 160×160 pixels to fit the 5-level pyramid and the block number in each level. The size of each image in the pyramid structure and the number of features is summarized in Table 6.5.

Pyramid level	Image Size	Descriptor Length
5	160×160	5200
4	128×128	3328
3	96×96	1972
2	64×64	832
1	32×32	108
Total		11440

Table 6.5: Pyramid structure of HOG descriptor.

6.3.3. LBP

As in the previous descriptor, a fixed block size of 32×32 is selected, so the original frames are scaled to 160×160 pixels. The size of each image in the pyramid structure and the number of features is summarized in Table 6.6.

Pyramid level	Image Size	Descriptor Length
5	160×160	1475
4	128×128	944
3	96×96	531
2	64×64	236
1	32×32	59
Total		3245

Table 6.6: Pyramid structure of LBP descriptor.

6.4. Feature Processing

6.4.1. PCA

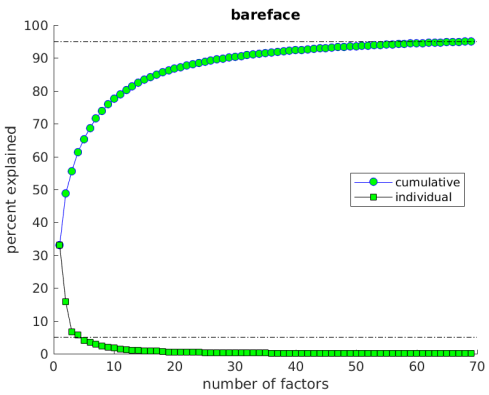
The first stage of feature processing is the reduction of the number of features using Principal Component Analysis. The initial number of features are shown in Tables 6.4, 6.5 and 6.6.

Initial study on principal components suggested that around a 60 % of variance can be explained with less than 10 features for the COV descriptor, less than 100 features for the HOG descriptor and less than 50 features for the LBP descriptor. However, preliminary tests reducing the dataset by the features that explain 60 % of the variance yielded results between 40 % and 60 % of accuracy, a result that is considered unacceptable.

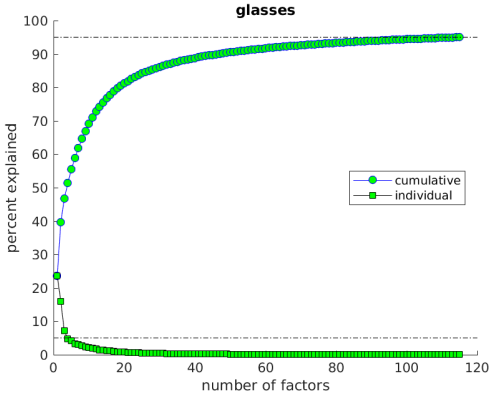
Considering the trade off between variance explained and the number of retained features, a value of 95 % of variance is set for COV and LBP descriptors and 85 % for HOG descriptor. In Figures 6.5, 6.6, and 6.7, the scree graph for the three descriptors has been plotted. The resulting number of features per descriptor and situation is shown in Table 6.7. It is noteworthy that for the PML-HOG descriptor reduced dataset, the number of features is significantly greater than for the other two descriptors, although the explained variance is 85 % vs 95 %.

Situation	PML-COV	PML-HOG	PML-LBP
bareface	69	487	123
glasses	115	643	171
sunglasses	139	670	184
night bareface	67	584	161
nightglasses	85	612	203

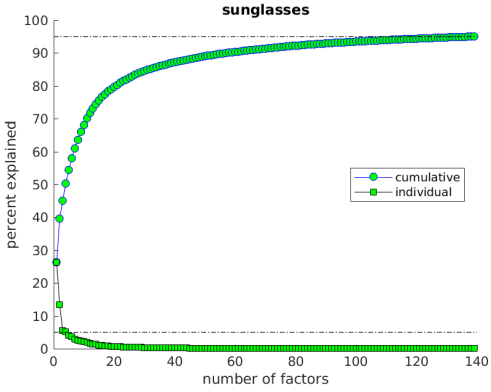
Table 6.7: Number of features after PCA for each descriptor and situation.



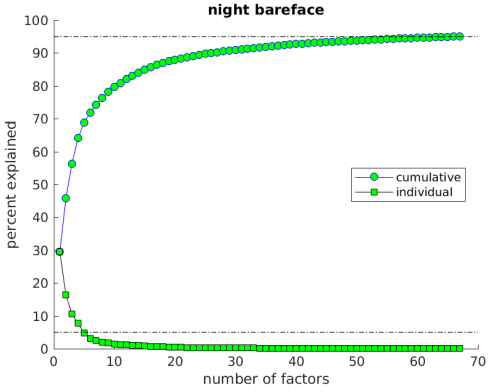
(a) bareface



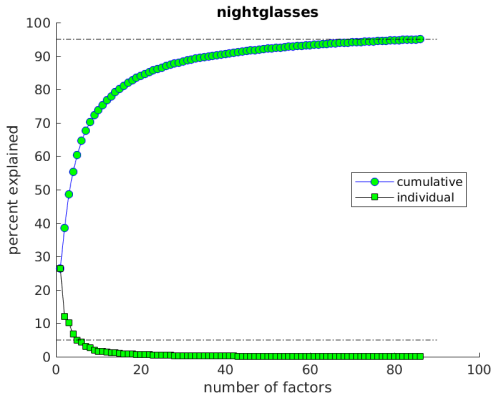
(b) glasses



(c) sunglasses

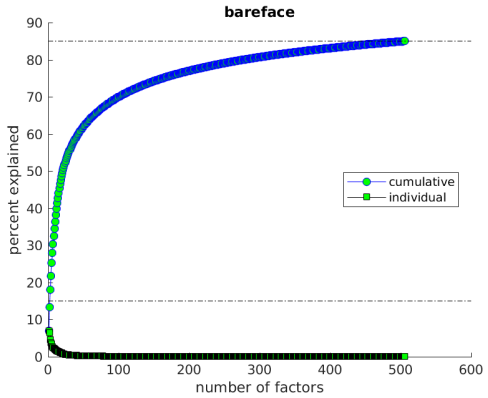


(d) night bareface

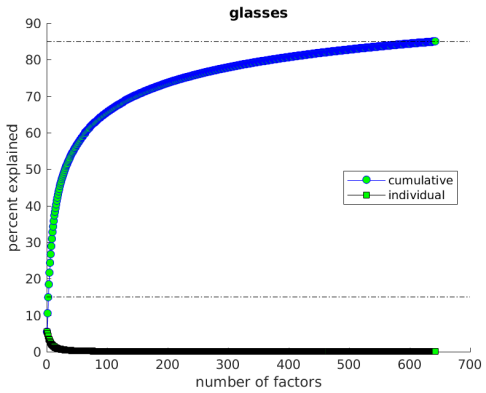


(e) nightglasses

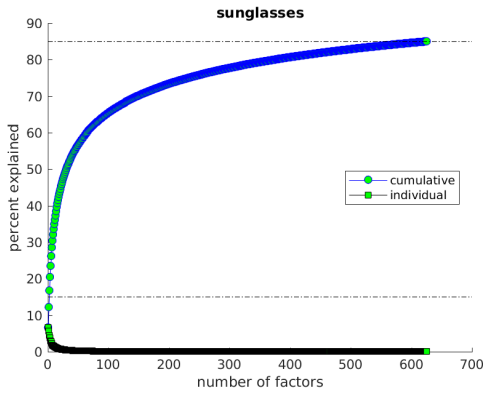
Figure 6.5: Scree graph of variance for COV descriptor by situation. Variance explained: 95 %.



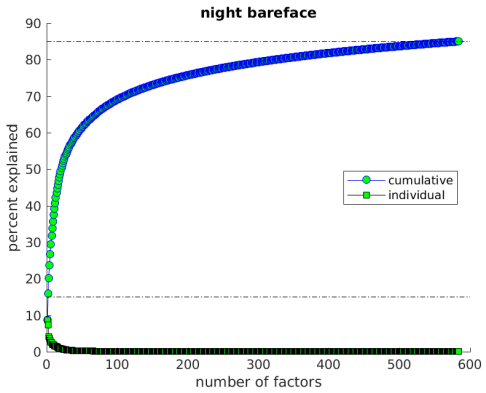
(a) bareface



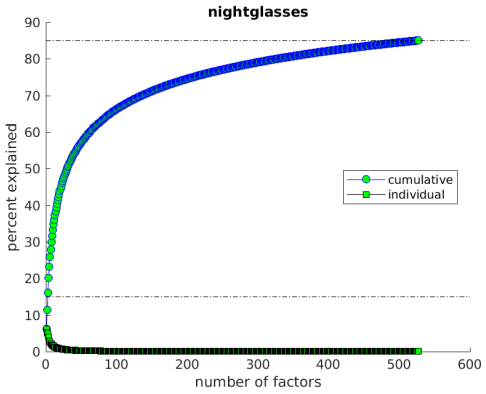
(b) glasses



(c) sunglasses

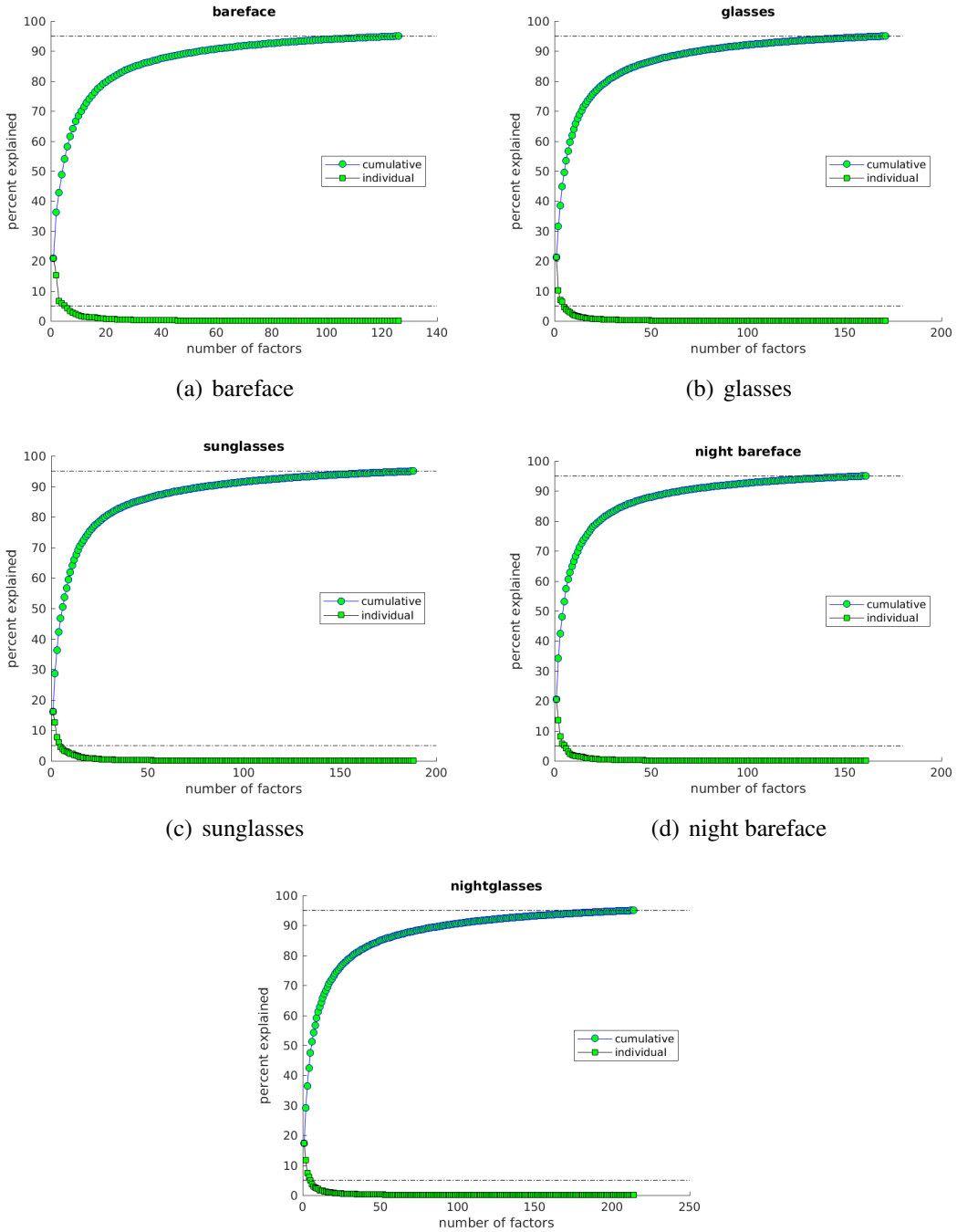


(d) night bareface



(e) nightglasses

Figure 6.6: Scree graph of variance for HOG descriptor by situation. Variance explained: 85 %.



(a) bareface

(b) glasses

(c) sunglasses

(d) night bareface

(e) nightglasses

Figure 6.7: Scree graph of variance for LBP descriptor by situation. Variance explained: 95 %.

6.4.2. Feature Selection

Feature selection has been performed according to Fisher's Score. The score allows to order a set of labeled features, from most important to less important, as explained in Section 3.2.

In order to extract the best set of features from a PCA-reduced dataset, the same pipeline has been applied selecting an increasing number of features each time and evaluating the pipeline accuracy for each round. Each round, the percentage of selected features is incremented and the final number of selected features is determined by the best accuracy obtained from the classifier.

As shown in Figures 6.8, 6.9 and 6.10 most situations with no glasses involved (*bareface* and *night bareface*) require less number of features, while *night glasses* and *glasses* need mid-level number of features to yield the best results. On the other hand, the *sunglasses* situation hardly benefits from feature selection as in most cases it requires nearly all available features to obtain the best results. This result may be produced by the difficulty to find out whether eyes are opened whenever the person is wearing sunglasses, which seems a crucial circumstance in the performance of the classifier.

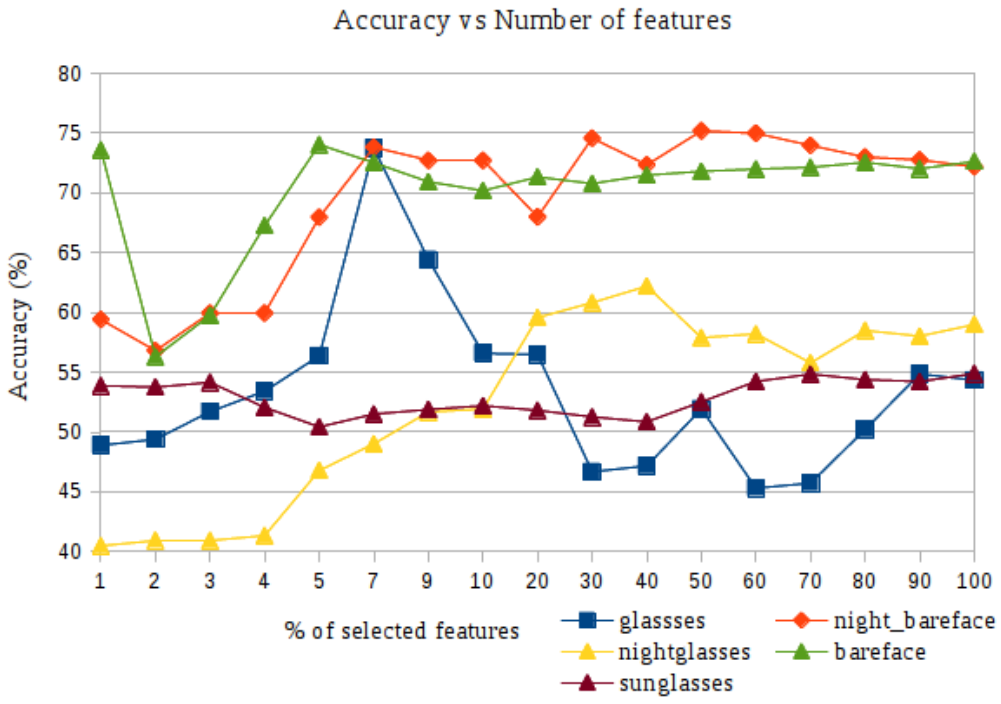


Figure 6.8: Classification results by selected number of features for COV descriptor.

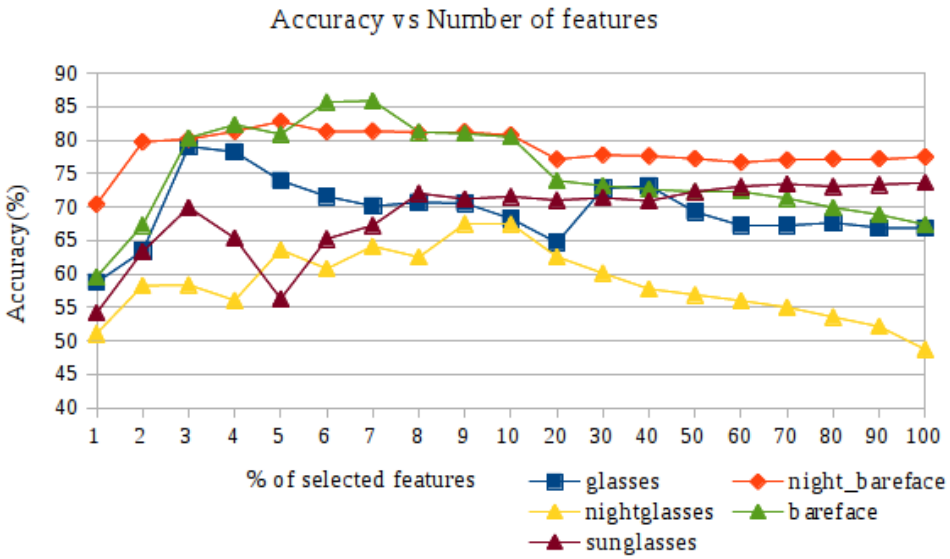


Figure 6.9: Classification results by selected number of features for HOG descriptor.

6.5. SVM parameters selection

Support Vector Machines require a few parameters to be adjusted. The first one is the regularization parameter, $\mathcal{C}$, which allows a certain amount of misclassification of non-linearly separable data. This parameter introduces a soft-margin, so the lower value of $\mathcal{C}$, the smaller penalty for 'outliers'. In this case, a RBF kernel has been used, so the other parameter to be optimized is the kernel σ.

The procedure followed to optimize both parameters consisted in a grid search. This search explores a discrete range of values of the aforementioned parameters. The accuracy of each trained classifier over the validation set is compared and the best accuracy value determines the parameters.

A first round is computed with bigger steps in the range of the possible values of parameters, and then a second round is computed with a more detailed value range. A representation of the first round is shown in Algorithm 6.1, where the same round is used to select the best set of features and the initial value of the σ value that will be fine-tuned in the second round along with the best value of $\mathcal{C}$. In this first round, the value of the $\mathcal{C}$ parameter is set to 1.

The second round is used to fine-tune the value of σ and the value of $\mathcal{C}$. In this case, a log-scaled range is used for $\mathcal{C} \in \{10^{-3}, 10^{3}\}$. The σ parameter is tuned by a grid search

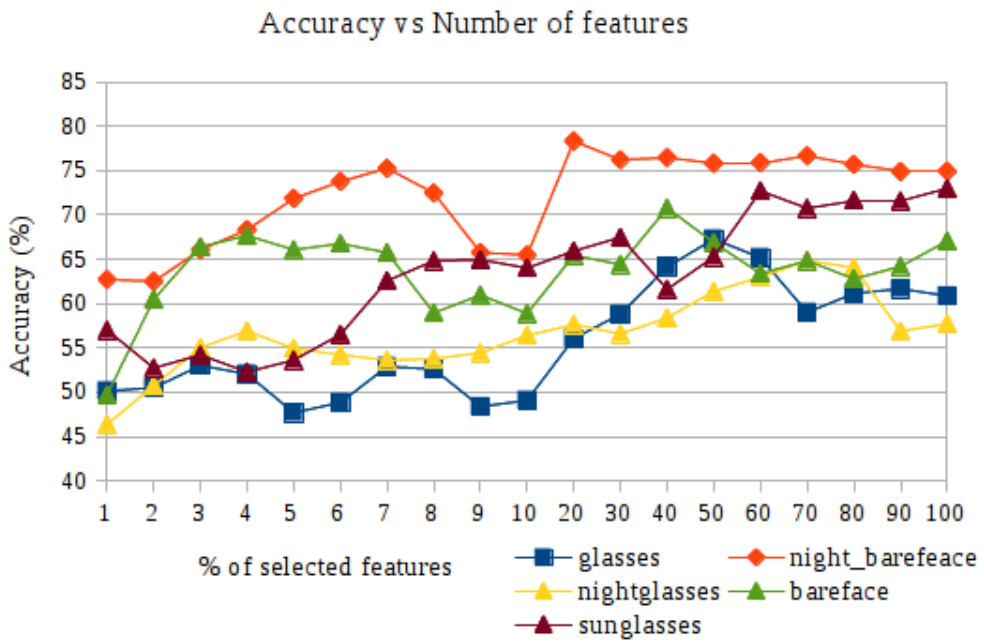


Figure 6.10: Classification results by selected number of features for LBP descriptor.

around the best valued of σ obtained in the first round with steps of 0.1. Algorithm 6.2 shows a illustrative diagram of the second round of the parameters estimation.

Algorithm 6.1: SVM parameters optimization (1st round)

Result: SVM raw estimation of parameter σ and number of features

begin

$bestAccuracy \leftarrow 0$
$best\_sigma \leftarrow 0$
$bestFeaturesSelected \leftarrow 0$
for $featuresSelected = 1\,\%$ **to** $100\,\%$ **do**
  $trainData \leftarrow \text{load}(trainingSetPCAReduced)$
  $trainData \leftarrow \text{featureSelection}(trainData, featuresSelected)$
  $validationData \leftarrow \text{load}(validationSetPCAReduced)$
  $validationData \leftarrow \text{featureSelection}(validationData, featuresSelected)$
  for $\sigma = 0.5$ **to** 50 **do**
    $(\sigma, accuracy) \leftarrow \text{FindBestClassifier}(trainData, validationData, \sigma)$
  if $accuracy \geq bestAccuracy$ **then**
    $best\_sigma \leftarrow \sigma$
    $bestFeaturesSelected \leftarrow featuresSelected$
return $(best\_sigma, bestFeaturesSelected)$

Algorithm 6.2: SVM parameters optimization (2nd round)

Result: SVM fine tuning of parameters σ and $\mathcal{C}$

begin

$bestAccuracy \leftarrow 0$
$best\_sigma \leftarrow 0$
$bestFeaturesSelected \leftarrow 0$
$(featureSet, sigma) \leftarrow \text{SVMParamsOptim1stRound}$
for $\mathcal{C} = 10^{-3}$ **to** 10^{3} **do**
  $trainData \leftarrow \text{featureSelection}(trainingSetPCAReduced, featureSet)$
  $validationData \leftarrow \text{featureSelection}(validationData, featuresSet)$
  for $\sigma = (sigma - 1)$ **to** $(sigma + 1)$ **do**
    $(\sigma, \mathcal{C}, accuracy) \leftarrow \text{FindBestClassifier}(trainData, validationData, \sigma, \mathcal{C})$
  if $accuracy \geq bestAccuracy$ **then**
    $best\_sigma \leftarrow \sigma$
    $bestC \leftarrow \mathcal{C}$
return $(best\_sigma, bestC)$

7.

Experimental results

This chapter shows the results obtained in the different experiments described in Chapter 5. Other important results are presented as the ability of generalization of different models, that is computed through cross-validation of the best models and of the pipeline itself.

As a first stage, an exploratory experimentation on the training set is carried out. For each subject, two groups of training sets can be used: the first group considers only two videos: drowsy state and non drowsy state, while the second group considers all available videos: drowsy state, non drowsy state, yawning and head nodding. At this stage, only the best feature set and the best σ value of the RBF kernel of SVM is optimized because the purpose of this analysis is to determine if there are differences in the results obtained with both datasets. In Tables 7.1, 7.2 and 7.3, the results on each dataset are used to select the best training set that will be used for further experiments.

Situation	2-videos dataset	4-videos dataset
bareface	75.93	81.00
glasses	65.82	73.81
sunglasses	64.24	56.90
night bareface	75,93	79.01
nightglasses	62.23	67.22

Table 7.1: Accuracy (%) by training set and situation for COV descriptor. Highlighted datasets correspond to best accuracy result.

Situation	2-videos dataset	4-videos dataset
bareface	75.79	86.20
glasses	77.81	79.07
sunglasses	73.68	72.62
night bareface	80.39	82.98
nightglasses	71.81	67.53

Table 7.2: Accuracy (%) by training set and situation for HOG descriptor. Highlighted datasets correspond to best accuracy result.

Situation	2-videos dataset	4-videos dataset
bareface	67.20	70.76
glasses	71.94	67.28
sunglasses	74.81	72.97
night bareface	62.77	78.31
nightglasses	62.13	64.84

Table 7.3: Accuracy (%) by training set and situation for LBP descriptor. Highlighted datasets correspond to best accuracy result.

7.1. Cross-validation

In Section 6.1 it is noted the lack of a true test set. The main purpose of this section is to compute the generalization abilities of the proposed approaches from Chapter 5.

The first approach to be tested is the one presented in Figure 5.1. Algorithm 7.1 is used to implement the cross-validation of the proposal, where the primary characteristic is that folds are built up making groups of subjects instead of grouping by samples.

The algorithm cross-validates 9 folds of 16 training subjects. In each fold, two subjects are extracted from the 18-subject training set and added to the 4-subject validation set, so statistical differences between training and validation set can be averaged over the 9 folds. This is done because statistical differences between training and validation sets can be influencing in the model performance. In each fold, the best classifier is built based on the procedure described in Algorithms 6.1 and 6.2 from Section 6.5. The best classifier for each fold is selected using the train and validation sets made for each fold.

Algorithm 7.1 is applied over each situation in the dataset and the results are displayed in Table 7.4. From this table, it can be derived that the descriptor PML-HOG yields the best generalization ability, although its value is close to the PML-LBP descriptor. The results

Algorithm 7.1: Cross-validation of a single situation

Result: Cross-validation accuracy

begin

$trainData \leftarrow$ load(training_set)
$validationData \leftarrow$ load(validation_set)
$subjectList \leftarrow$ list of training subjects
$AccuracyList \leftarrow emptyList$
for $fold = 1$ **to** $length(subjectList)/2$ **do**
  $subjects \leftarrow$ selectWithoutRepetition$(subjectList, 2)$
  $extractedData \leftarrow$ extractSubject$(trainData, subjects)$
  $foldTrainData \leftarrow$ removeData$(trainData, extractedData)$
  $foldValidationData \leftarrow$ joinData$(validationData, extractedData)$
  $Model \leftarrow$ FindBestClassifier$(foldTrainData, foldValidationData)$
  $accuracy \leftarrow$ Accuracy$(Model, foldValidationData, foldValidationLabels)$
  Append$(AccuracyList, accuracy)$
return Average$(AccuracyList)$

for PML-COV are relatively lower compared to the other two descriptors.

Situation	PML-COV	PML-HOG	PML-LBP
bareface	77.53	80.37	72.59
glasses	67.08	75.88	73.02
sunglasses	65.84	73.74	77.89
night bareface	77.53	80.37	76.68
nightglasses	66.12	67.55	67.13
Average	70.82	75.58	73.46

Table 7.4: Cross-validation detection accuracy results (%) applying Algorithm 7.1 to each situation in the dataset.

Algorithms 6.1 and 6.2 are used to optimize the SVM parameters and select the best feature set. Cross-validation for the optimized SVM parameters C and σ and for the best set of selected features are also computed for every situation. The cross-validation is computed with a variation of Algorithm 7.1 where, instead of trying to find the best classifier, a SVM model with optimized C and σ parameters is trained.

7.2. Pipeline results

The parameters optimization for every proposed approach from Chapter 5 are summarized in Table 7.5. As individual descriptors, PML-HOG performs best while PML-COV and PML-LBP have similar results. The combination of pipelines for the three image descriptors, defined here as SVM blending, shows the best result with a 79.84 % accuracy in drowsiness detection. The results after concatenating the three descriptors are higher than the average of the three descriptors, but it does not improve the best results on individual descriptors nor SVM blending.

The difficulty to detect the eyes under the circumstances where any kind of glasses are involved highly penalizes the results, specially when the subject is wearing sunglasses because the light is reflected on the surface of the glasses.

Situation	bareface	glasses	sunglasses	night bareface	night-glasses	Average
PML-COV	82.34	74.19	67.19	79.76	68.90	**74.48**
PML-HOG	87.02	79.10	73.97	83.88	73.10	**79.41**
PML-LBP	73.58	74.04	77.27	78.38	71.00	**74.85**
Blending SVM	85.76	78.31	76.86	83.76	74.52	**79.84**
Concatenate descriptors	80.28	76.59	70.76	82.30	73.94	**76.77**
Concatenate reduced descriptors	81.21	76.22	72.40	84.60	70.17	**76.92**

Table 7.5: Detection accuracy results (%) on validation set.

If one subject from the original validation set is excluded from the pipeline, then this data can be used as a test set. Repeating this process for every subject in the original validation set and averaging the accuracy results, allows to estimate the performance of the pipeline on new data.

This concept is applied for each one of the three descriptors and the pipeline proposed in Section 5.1 to simulate a pure process with training, validation and test sets. As there are four available validation subjects, four classifiers are trained and optimized using only training and validation data and they are tested on the extracted subject. Table 7.6 shows the results for this experiment where it is clear that the performance of the models drop drastically. However, PML-HOG still remains as the best image descriptor.

Situation	PML-COV	PML-HOG	PML-LBP
bareface	66.84	70.55	72.17
glasses	52.39	67.51	65.96
sunglasses	58.49	62.63	55.13
night bareface	70.95	76.52	70.84
nightglasses	45.69	72.64	66.35
Average	58.87	69.97	66.09

Table 7.6: Detection accuracy results (%) averaged for experiments on 4 different test sets using one validation subject as test set on each experiment.

The results can be compared with other works from state-of-the-art papers on the same data set. These results are summarized in Table 7.7. Results for models that are best fitted for the validation set (Table 7.5) outperform previous works. Results for models evaluated on a pure test set (Table 7.6) are comparable to other works in the best case (PML-HOG).

A more detailed analysis of the results prove that one particular subject from the validation set offers much lower results than the other subjects and that issue may have influence on the final results. In Figure 7.1 best results on the validation set for each image descriptor (Table 7.5) are separated by validation subject. It can be seen that the average performance of subject *004* is highly affected by the glasses situation.

In the case where a subject from the original validation set is extracted from the validation set and used as a test set, the effect is similar for subject *004*. In Figure 7.2, the results when each subject is used as a test set are shown. Each column represents the test results when the subject is used as a test set and the other three subjects as validation set.

PML-COV	Subject			
Situation	4	22	26	30
glasses	9.20	97.94	75.33	82.89
night bareface	79.71	89.27	74.56	59.69
nightglasses	73.28	63.86	75.50	68.93
bareface	86.85	89.45	75.51	58.36
sunglasses	68.47	98.29	63.61	62.47
Average	63.50	87.76	72.90	66.47

PML-HOG	Subject			
Situation	4	22	26	30
glasses	54.70	79.88	76.91	86.79
night bareface	85.18	84.28	80.88	83.26
nightglasses	56.49	65.84	98.12	78.44
bareface	83.67	95.31	84.53	70.68
sunglasses	69.22	91.33	68.87	78.31
Average	69.85	83.33	81.86	79.49

PML-LBP	Subject			
Situation	4	22	26	30
glasses	33.44	99.72	76.33	75.75
night bareface	83.34	74.93	69.29	86.18
nightglasses	86.48	44.99	81.09	77.22
bareface	66.61	79.66	76.61	60.16
sunglasses	75.38	100.00	65.65	85.49
Average	69.05	79.86	73.79	76.96

Figure 7.1: Accuracy (%) per validation subject when the 4 subjects are used as validation set and SVM is trained with the training set and its parameters optimized with the validation set.

Situation	LRCN [2]	DDD-FFA [3]	DDD-IA [3]	3D DCCN [4]	PML COV	PML HOG	PML LBP	Blending SVM	Concatenate Descriptors	Concatenate Reduced Descriptors
bareface	68.75	79.41	69.83	75.1	82.34	**87.02**	73.58	85.76	80.28	81.21
glasses	61.73	74.10	75.93	72.3	74.19	**79.10**	74.04	78.31	76.59	76.22
sunglasses	71.47	61.89	69.86	70.9	67.19	73.97	**77.27**	76.86	70.76	72.40
night bareface	57.39	70.27	74.93	68.4	79.76	83.88	78.38	83.76	82.30	**84.60**
nightglasses	55.63	68.37	**74.77**	68.3	68.90	73.10	71.00	74.52	73.94	70.17
Average	62.99	70.81	73.06	71.2	74.48	79.41	74.85	**79.84**	76.77	76.92

Table 7.7: Detection accuracy results (%) from [2] (LRCN), [3] (DDD-FFA and DDD-IA), [4] (3D-DCCN) and ours. In bold the best results for each situation.

PML-COV	Subject			
Situation	4	22	26	30
glasses	9,00	74,83	58,06	67,65
night bareface	66,65	89,25	68,30	59,61
nightglasses	64,09	45,70	19,08	53,88
bareface	75,48	68,82	66,61	56,47
sunglasses	68,78	98,29	33,65	33,24
Average	56,80	75,38	49,14	54,17

PML-HOG	Subject			
Situation	4	22	26	30
glasses	36,02	76,42	73,87	83,76
night bareface	77,94	69,89	76,11	82,06
nightglasses	86,20	61,64	72,51	70,39
bareface	79,10	61,75	70,78	70,68
sunglasses	71,20	36,64	66,86	75,84
Average	70,09	61,27	72,03	76,55

PML-LBP	Subject			
Situation	4	22	26	30
glasses	33,42	99,76	57,58	73,07
night bareface	71,35	56,65	69,19	86,18
nightglasses	85,48	44,02	80,27	55,64
bareface	40,32	53,47	63,97	62,74
sunglasses	57,13	84,95	72,80	73,78
Average	57,54	67,77	68,76	70,28

Figure 7.2: Accuracy (%) when one subject is used as test set and the other three as validation set. Results in each column correspond to the test set accuracy when that particular column is the test set.

8.

Conclusions

This work proposes and compares 4 pipelines for detecting drowsiness and fatigue in drivers. Three different image descriptors have been considered to extract image features from subjects with different age, gender and ethnicity. Another fourth descriptor is built as the concatenation of the three aforementioned image descriptors.

The analysis of the results shows that models trained using only one image descriptor can yield very good results in the case of PML-HOG descriptor while PML-COV and PML-LBP cannot be compared to the HOG descriptor results. The fact that drowsiness is mainly expressed by the shape of facial attributes may suggest that PML-HOG yields best performance since this descriptor is best suited to capture shapes, corners and edges. The concatenation of descriptors to form a new high dimensional descriptor does not improve the best result for and individual descriptor and besides the computational cost is to high to consider this descriptor over the individual HOG descriptor. On the other hand, a blending SVM model where three independent pipelines are executed in parallel and the final classification is computed as the average of each individual pipeline yields the best result.

Comparison against work of other authors on the same dataset reveals that an improvement on performance of the proposed pipelines has been achieved.

Future work should include a different classification algorithm such as convolutional neural networks(CNN), although CNN could also be used as an image descriptor if the final layer is used instead of the CNN classification result.

9.

Appendix

The code is available on GitHub in the following link so that anyone can replicate the results of this work: https://github.com/jretac/Driver-Drowsiness-Detection.

Bibliography

[1] K. Zhang, F. Zhang, J. Lu, Y. Lu, J. Kong, and M. Zhang, "Local structure co-occurrence pattern for image retrieval," *Journal of Electronic Imaging*, vol. 25, no. 2, p. 023030, apr 2016.

[2] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *CoRR*, vol. abs/1411.4389, 2014.

[3] S. Park, F. Pan, S. Kang, and C. D. Yoo, "Driver Drowsiness Detection System Based on Feature Representation Learning Using Various Deep Networks," *Proceedings of the 7th International Conference on Advances in Pattern Recognition*, vol. 44, no. 4, pp. 426–429, mar 2009.

[4] J. Yu, S. Park, S. Lee, and M. Jeon, "Representation learning, scene understanding, and feature fusion for drowsiness detection," in *Computer Vision – ACCV 2016 Workshops*. Springer International Publishing, 2017, pp. 165–177.

[5] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 596–614, June 2011.

[6] T. Åkerstedt and M. Gillberg, "Subjective and objective sleepiness in the active individual," *International Journal of Neuroscience*, vol. 52, no. 1-2, pp. 29–37, 1990.

[7] Y. Takei and Y. Furukawa, "Estimate of driver's fatigue through steering motion," *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, no. 1, pp. 1–6, 2005.

[8] G. Niu and C. Wang, "Driver Fatigue Features Extraction," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–10, 2014.

[9] B. T. Jap, S. Lal, P. Fischer, and E. Bekiaris, "Using EEG spectral components to assess algorithms for detecting fatigue," *Expert Systems with Applications*, vol. 36, no. 2 PART 1, pp. 2352–2359, 2009.

[10] M. H. Sigari, "Driver hypo-vigilance detection based on eyelid behavior," in *Proceedings of the 7th International Conference on Advances in Pattern Recognition, ICAPR 2009*. IEEE, feb 2009, pp. 426–429.

[11] M. Singh and G. Kaur, "Drowsy Detection On Eye Blink Duration Using Algorithm," *International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com*, vol. 2, no. 4, pp. 363–365, 2012.

[12] K. W. Kim, H. G. Hong, G. P. Nam, and K. R. Park, "A study of deep CNN-based classification of open and closed eyes using a visible light camera sensor," *Sensors (Switzerland)*, vol. 17, no. 7, 2017.

[13] N. Kumar and N. C. Barwar, "Detection of Eye Blinking and Yawning for Monitoring Driver ' s Drowsiness in Real Time," *International Journal of Application or Innovation in Engineering & Management*, vol. 3, no. 11, pp. 291–298, 2014.

[14] C. Weng, Y. Lai, and S. Lai, "Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network," in *Asian Conference on Computer Vision - Workshop on Driver Drowsiness Detection from Video*, Taipei, Taiwan, Nov2016.

[15] J. Lyu, Z. Yuan, and D. Chen, "Long-term multi-granularity deep framework for driver drowsiness detection," *CoRR*, vol. abs/1801.02325, 2018.

[16] A. Dasgupta, A. George, S. L. Happy, and A. Routray, "A vision-based system for monitoring the loss of attention in automotive drivers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1825–1838, Dec 2013.

[17] I. Teyeb, O. Jemai, M. Zaied, and C. Ben Amar, "A drowsy driver detection system based on a new method of head posture estimation," in *Intelligent Data Engineering and Automated Learning – IDEAL 2014*. Cham: Springer International Publishing, 2014, pp. 362–369.

[18] N. Alioua, A. Amine, and M. Rziza, "Driver's fatigue detection based on yawning extraction," in *Int. J. Veh. Technol*, Aug 2014.

[19] I.-H. Choi, S. K. Hong, and Y.-G. Kim, "Real-time categorization of driver's gaze zone using the deep learning techniques," in *2016 International Conference on Big Data and Smart Computing (BigComp)*, Jan 2016, pp. 143–148.

[20] S. R. Arashloo and J. Kittler, "Dynamic texture recognition using multiscale binarized statistical image features," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2099–2109, Dec 2014.

[21] S. E. Bekhouche, A. Ouafi, F. Dornaika, A. Taleb-Ahmed, and A. Hadid, "Pyramid multi-level features for facial demographic estimation," *Expert Systems with Applications*, vol. 80, pp. 297–310, 2017.

[22] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Computer Vision – ECCV 2006*. Springer Berlin Heidelberg, 2006, pp. 589–600.

[23] "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[24] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[25] "R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, New York: John Wiley & Sons, 2001, pp. 654, ISBN: 0-471-05669-3," *Journal of Classification*, vol. 24, no. 2, pp. 305–307, sep 2007.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, sep 1995.

[27] R. Fletcher, *Practical Methods of Optimization; (2Nd Ed.)*. New York, NY, USA: Wiley-Interscience, 1987.

[28] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874, 2014.

[29] D. G. Lowe, "Distinctive image features from scale invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–11 020 042, 2004.