FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Passenger identification and monitoring using thermal images

Rui Paulo Araújo Monteiro

WORKING VERSION

MASTER THESIS

Supervisor: Luís Côrte-Real

Second Supervisor: Pedro Carvalho

External Supervisor: André Ferreira

August 1, 2018

Abstract

With the latest breakthroughs in autonomous driving, we are getting closer to a future where shared autonomous vehicles will be common. As such, solutions should be developed in order to carry out the tasks that are currently done by the driver in transportation systems, such as car cleaning and maintenance and generally ensuring that everything is okay with the passengers aboard. Even though there are already some solutions developed using visible images, they are not robust to all the different conditions that we may encounter inside the vehicle during the 24 hours of the day (different light conditions, for example).

Many papers were written on using thermal imaging for the tasks of face detection, face recognition and landmark detection. However, they were not designed for in-vehicle purposes. With this in mind, the main objective of this thesis is the development of algorithms specifically designed to be able to accomplish the tasks of face detection, facial keypoint detection, face recognition, fatigue and drowsiness monitoring and temperature monitoring using thermal images taken from inside the vehicle.

To accomplish this, a dataset containing a total of 38 subjects was created, with time synchronized RGB and thermal images. For face detection, transfer learning was applied with YOLOv3 to train a neural network capable of achieving around 79% mean Average Precision (mAP) in our dataset. A siamese neural network was trained for the task of face recognition, able to achieve around 97 % top-1 accuracy in the dataset that we created, and for fatigue monitoring an eye state classifier (opened or closed) was developed using thermal images, able to achieve around 91% accuracy on the validation set.

Resumo

Com os recentes desenvolvimentos no âmbito da condução autónoma, estamos cada vez mais próximos de um futuro onde veículos autónomos partilhados serão comuns. Assim, devem ser desenvolvidas soluções para substituir a função do motorista nos sistemas de transporte actuais, tais como manutenção e limpeza do carro e, no geral, garantir que tudo está bem com os passageiros. Apesar de já existirem algumas soluções capazes desenvolvidas para este propósito em imagens na gama do visível, estas não são robustas em relação a todas as diferentes condições que poderemos encontrar num carro durante o dia (diferentes condições de luminosidade, por exemplo).

Diversos artigos foram escritos sobre usar imagens térmicas para tarefas como deteção facial, reconhecimento facial e deteção de pontos fiduciais. No entanto, não foram desenvolvidos para este propósito em específico. Tendo isto em conta, o objetivo principal desta dissertação é o desenvolvimento de algoritmos capazes de realizar deteção facial, deteção de pontos fiduciais em faces, reconhecimento facial, deteção de fadiga e sonolência e também monitorização de temperatura com imagen térmicas tiradas dentro do veículo .

Para o efeito, um dataset contendo um total de 38 pessoas foi criado, com imagens térmicas e RGB sincronizadas no tempo. Na deteção facial, foi usado a técnica de transfer learning com a rede neuronal convolucional YOLOv3 para treinar uma rede capaz de atingir à volta de 79% de precisão de mAP. Uma rede neuronal siamesa foi treinada para verificação facial, capaz de atingir cerca de 97 % de precisão top-1 num dataset contendo 79 pessoas, e para deteção de fadiga e sonolência um classificador de olhos (fechado ou aberto) foi desenvolvido usando imagens térmicas, capaz de atingir à volta de 91 % de precisão nas imagens de validação.

Acknowledgments

After an intensive period of five months, today is the day: writing this note of thanks is the finishing touch on my dissertation. It has been a period of intense learning for me, not only in the scientific arena, but also on a personal level. Writing this dissertation has had a big impact on me. I would like to reflect on the people who have supported and helped me so much throughout this period.

I would first like to thank my colleagues from my internship at Bosch Car Multimedia, Gustavo, Guilherme and Tiago Fernandes, for their wonderful collaboration. You supported me greatly and were always willing to help me.

In addition, I would like to thank my tutors, Luís Côrte-Real and Pedro Carvalho from INESC and also André Ferreira and Joaquim Fonseca from Bosch Car Multimedia, for their valuable guidance. You definitely provided me with the tools that I needed to choose the right direction and successfully complete my dissertation.

I would also like to thank my parents for their wise counsel and sympathetic ear. You are always there for me. Finally, there are my friends. We were not only able to support each other by deliberating over our problems and findings, but also happily by talking about things other than just our papers.

Thank you very much, everyone!

Rui Monteiro

"The pessimist sees difficulty in every opportunity.
The optimist sees opportunity in every difficulty."

Winston Churchill

Contents

List of Figures

List of Tables

Abbreviations

AP	Average Precision (same as mAP)
CNN	Convolutional neural network
EAR	Eye aspect ratio
FOV	Field-of-view
FPS	Frames per second
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
MSE	Mean squared error
RGB	Red-Green-Blue
SARS	Severe acute respiratory syndrome
SVM	Support vector machine
U.S.	United States
USB	Universal Serial Bus
T	Thermal

Chapter 1

Introduction

1.1 Context

With the latest breakthroughs and developments in self-driving cars, the human race is getting closer to a future where these vehicles can be independent: a person enters the car, picks her target address and comfortably sits there while the car drives her to the target location. This allows for the possibility of car sharing, where no one owns a vehicle, and every vehicle on the road is shared by everyone (Shared Autonomous Vehicles – SAV).

In modern days, where a lot of the effort is focused on solving and perfecting the task of autonomous driving, this thesis is more concerned with the issues and options inside the vehicle, specifically related to the comfort and interaction with people inside the car.

Even though several breakthroughs were made in the visible modality, where pratically everything is well established (joint detection, person detection and recognition, activity classification, head pose estimation and visual gaze), these images depend on external conditions, namely light, so because of this they are not robust to every situation that we may encounter during the 24 hours of a day. With this in mind, several modalities are being explored, that can be used independently or in conjunction, namely near infra-red (NIR), robust to lack of lightning but needs a dedicated source of NIR light and filters and it's sensible to different lens exposures, and the modality that we will focus on, thermal images, whose principle of functioning is in the thermal radiance emitted by the objects, and therefore robust to every external light situation that we may encounter.

1.2 Motivation

Nowadays, several accidents occur on the road because of human errors. According to [6], 94% of car accidents are caused by this. In this regard, autonomous driving can be an effective solution to prevent road accidents that are responsible for causing thousands of deaths per year worldwide [7].

In accordance with what was said before, new questions and problems may arise, related to the comfort and safety of the passengers inside the vehicle. In the context of shared autonomous

vehicles, these types of services are expected to be used by everyone, so it is crucial that solutions are developed to control and prevent events such as, for example, the widespread of infections diseases, counting and detecting who exactly is inside the car at a given moment , detecting if a person is sleeping or detecting if someone is smoking inside the vehicle.

Analyzing the state of the art in thermal images, we noticed that the common approaches for the mentioned tasks were slightly outdated. With the recent breakthroughs and developments achieved in the visible (RGB) domain, we wanted to try and implement these techniques specifically to thermal images, to check if they too can achieve good results in different types of images.

1.3 Objectives

This thesis was proposed by Bosch Car Multimedia S.A. and the main objective of this thesis is the development and solutions to be applied inside the car, related with information about people inside the car and their fitness status, adopting the usage of thermal cameras or a combination of thermal and RGB cameras (RGB-T).

To do so, the following tasks are considered:

1. Dataset creation: to develop and form these new algorithms, and due to the lack of good datasets that included RGB and thermal images, we felt the need to create a new one and custom-made to our purposes. Working with a big company such as Bosch Car Multimedia S.A., we had favorable circumstances to gather a large group of people to compose the dataset.

2. Face Detection: implement algorithms capable of detecting all the faces present inside the vehicle, while being robust to different head poses, facial expressions and overall faces (regarding differences in hair, beards and wearing glasses). This can be important for people counting and as a first step for other tasks such as face recognition and facial keypoint estimation.

3. Facial Landmark Detection: develop methods able to perform detection of the position of facial keypoints, like eyes, nose and mouth. This can be important for other tasks such as head pose estimation, face alignment and facial expression recognition

4. Temperature Monitoring: develop an algorithm capable of estimating a person's body temperature using a thermal camera. This can be important for fitness-related profiling.

5. Face Recognition: implement an algorithm capable of detecting who is the person inside the vehicle and if the person is allowed inside.

6. Fatigue and Drowsiness Monitoring: detecting if a person is tired or if the person is sleeping or in a state of almost sleeping.

It's worth mentioning that, because these kinds of applications (excluding body temperature estimation) are well researched in the visible domain, we aim to develop algorithms that are robust to all the different conditions that we may encounter inside the car throughout the day, such as different light conditions, variable temperature conditions and different "face" conditions (for example using glasses, having long beards, long hair ...).

1.4 Contributions

This thesis includes several contributions to the thermal community, namely:

1. A visible and thermal dataset was created, with a total of 38 persons. The dataset is labelled and includes images taken inside a vehicle, with subjects performing several different activities such as head movements, facial expressions and fatigue and drowsiness simulations.

2. A face detector was developed using thermal images, robust to different conditions such as difficult head poses and environmental temperatures

3. A facial keypoint detector was made using thermal images, capable of detecting several landmarks including eye points, mouth and nose.

4. An algorithm was developed able to estimate a person's body temperature based on the eye corner temperatures.

5. A thermal face recognizer was created using a state-of-the-art siamese neural network, able to distinguish between persons that are inside a database

6. A convolutional neural network capable of detecting if a person has her eyes opened or closed using thermal images

+

Chapter 2

Related Work

In this section, we give an introduction to thermal images, how they work and some current applications of this type of images. Also, some background will be provided on the general state of the art and techniques related to ordinary object detection, face detection, face recognition, head pose estimation and fatigue and drowsiness detection, using visible (RGB) images and thermal images. We'll start with some background theory related to this type of work followed by the current state of the art.

2.1 Thermal Images

Thermal cameras operate in the Long-wavelength infrared (LW-IR) wavelengths, with range from 8 µm-12 µm. Because they operate in this region of the spectrum, they are able to detect electromagnetic radiation emitted from all surfaces with a temperature above absolute zero (0K = -273,15ºC). The camera operation principle is based on the Stefan-Boltzmann law, which states that the amount of IR radiation emitted by a body is proportional to its emissivity (e), its temperature and the Stefan-Boltzmann constant ($\sigma = 5,6697x10^{-}8$)[8].

Emissivity can be defined as the ratio of thermal radiation emitted from a surface compared to the radiation emitted from an ideal black body (e = 1) at the same temperature and with the same area[9]. A surface with an emissivity of zero does not absorb nor emit energy, and a surface with an emissivity of one absorbs all energy. Relating this information to our case, the emissivity of the human body is close to one (ε = 0.980$\pm$0.01) [10], so we are able to estimate its surface temperature using this type of image.

Recently, thermal imaging is gaining more importance in scientific research, mainly because of technological advances in uncooled thermal detectors made by the United States (U.S.) military[8]. This means that nowadays, we can see a significant price drop on these types of cameras. In modern uncooled thermal cameras, infrared radiation is detected by focal plane arrays (FPA) of vanadium or silicon microbolometers[11]. These microbolometers are responsible for capturing thermal information of a pixel. To produce the image, the data from each microbolometer is combined into a pseudo-color image, like the ones we see in Figure 2.1.

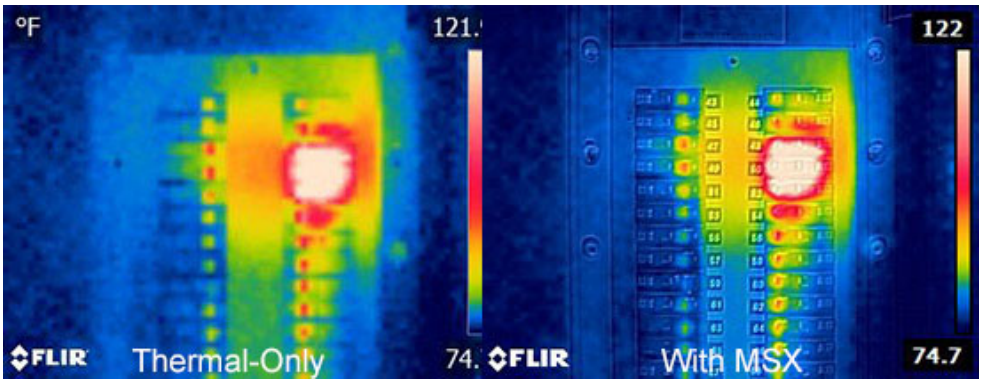

Figure 2.1: Thermal-Only and MSX comparison, taken from FLIR website

Similarly to visible light cameras, lenses are used in thermal cameras to focus the infrared radiation onto the focal plane array, although made of different materials (germanium crystals and chalcogenide glass) [8]. These types of cameras normally have low spatial resolution. This has the effect of averaging temperatures across pixels, therefore introducing some error[9]. One way to tackle this issue is by combining visible light images with thermal images: since visible light cameras are available in higher resolutions, the images can be combined (RGB and Thermal (T) data fusion) to produce a higher quality image with thermal information. Also, we can keep segments of each type of images to accomplish different purposes independently. One example of RGB-T fusion is the MSX technology, available in most FLIR cameras. An image of this technology is presented in Figure 2.1.

2.1.1 Body Temperature Estimation

In humans, normal body temperature ranges from 36.5°C to 37.5°C, and a higher value than this range may suggest that a person has fever. Basically, fever occurs when the human body decides to raise its core body temperature to help fight some disease: in other words, fever can be a sign of infection. It can be a simple and non-serious one, or it can be an indicator of more complicated diseases. Fever screening systems, such as the ones we see in some airports, can be an effective way of preventing the widespread of infectious viruses and diseases such as Ebola, Severe acute respiratory syndrome (SARS) and Tuberculosis[10].

Numerous studies have attempted to automatically detect human body temperature through infrared thermography (thermal images), and have shown that there is a high correlation between the temperature measured by a regular thermomether in the axilla and the value measured in a thermogram on the inner canthi of a person's eye[12][13], which is the corner of the eye. For this detection to be accurate, the image should contain all of the face of the person (implying a robust process of face detection) and the thermal camera should be calibrated properly, to guarantee an accurate measure of temperature[14].

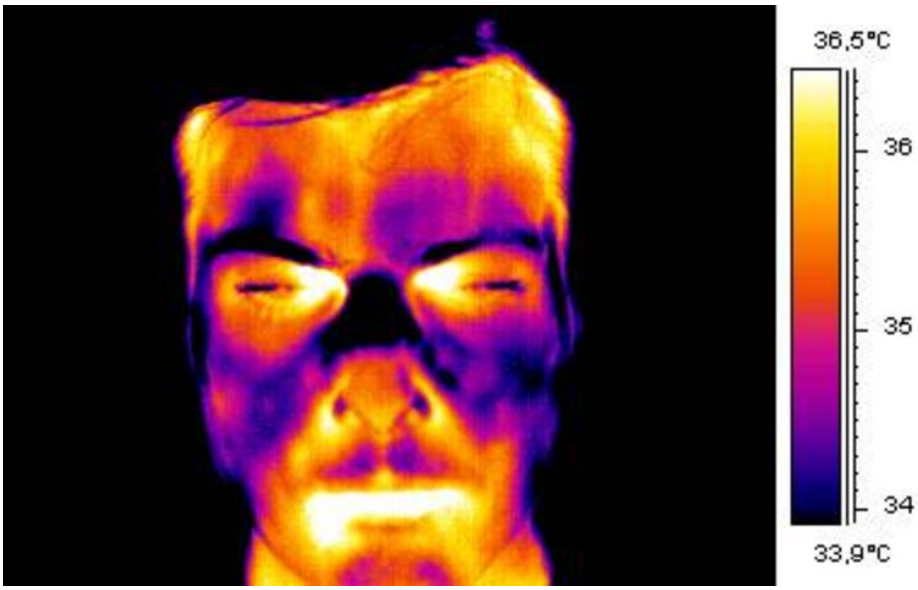

Figure 2.2: Thermal Image from [1]. There is an intensity-maximum region in the inner corner of the eye.

2.1.2 Thermal Image Preprocessing

In thermal images, the pre-processing step is a crucial one. Roughly speaking, thermal images offer low resolutions and because their color/brightness information is based on the temperature of the object and the environment, they can suffer from low contrast. With these issues in mind, it is important to compensate for these effects, by using image processing procedures.

Usually, for object detection in thermal images, objects may be easier to detect after applying Otsu's[15] method or an adaptative threshold method to process the image. Otsu's method tries to separate and group pixels into two classes, by minimizing their intra-class variance and maximizing their inter-class variance[15]. Generally, with some adaptation this can result in separating the foreground from the background. Another commonly used method, employed by the authors in [16] is the gradient method for edge detection. Implementing an algorithm like Canny[17] can be helpful to detect significant contours that are useful for object description and detection. Lastly, other algorithm that can be applied is DOG (difference of gaussian)[18], which can be used mainly in local feature extraction to remove unwanted noise and emphasize the required features (mainly edges).

In computer vision, low contrast in images are frequently corrected by using histogram equalization methods. In [19], the authors suggest that for thermal images it is better to use CLAHE (contrast limited adaptive histogram equalization) instead of normal histogram equalization, and their simulation results show that this algorithm can enhance their visual quality and peak-to-peak signal-to-noise ratio (PSNR). A comparison can be seen in Figure 2.3.

Figure 2.3: a) Thermal Image (taken from public IRIS dataset) b) Histogram-equalized image c) CLAHE image. It can be noticed that b) still suffers from low contrast, while c) is a much higher contrast image

2.2 Object Detection

Object detection is the process of finding instances of objects (persons, dogs, cars, faces, ...) in images or videos. Like object detection, there are some other similar and related tasks, such as image classification, the task of assigning one label from a fixed number of predetermined categories to a given image, and also semantic segmentation, to partition the image into meaningful parts by classifying and labeling each pixel of the input image [20].

For many years, the traditional approach to solve the problem of general object detection was to extract exclusive features from objects that would be used as an input to a classifier, to be able to assign a label to the objects. In its simple form, features can be something simple as height, width, form or color, while some of the most used and complex features include Haar features[21], SIFT[22] and SURF[23] and HOG features[24].

After the process of feature extraction, these would later be classified through algorithms such as simple thresholds, Support Vector Machine (SVM) or boosted classifiers on sliding windows. However, using these kinds of methods is difficult because they have to take into account lots of issues that are hard and unpredictable to solve, like differences in viewpoints, different light conditions and different resolutions[25]. It's very difficult and time consuming to design an algorithm like this that is capable of solving these different kinds of inherent issues.

Nowadays, the approach in modern object detection is the implementation of deep convolutional neural networks (CNN). In contrast to traditional computer vision approaches, deep learning methods avoid the hand-crafted feature design procedure and have solved many known benchmark datasets, such as ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [26].

2.2.1 Related Work in Object Detection

It was not until 2012 that we would see a huge development in object detection using CNN's. AlexNet [27] was able to win the 2012 ILSVRC(ImageNet Large-Scale Visual Recognition Challenge). Their architecture was able to achieve a top-5 test error rate of 15.4%, with a huge difference to the team that took second place who accomplished an error of 26.2%. The network used five convolutional and max-pooling layers and also dropout layers, a regularization technique used

to prevent adaptation on training data, with the purpose of reducing overfitting. Also, data augmentation techniques like horizontal reflections and patch extractions were used in order to create new images for the training of the network. This paper was the first one to demonstrate the benefits of implementing CNN's for the task of object detection: even though they were previously used [28], the current advancements in CPU's/GPU's and also the appearance of large datasets stimulated their reappearance.

Afterwards, CNN's became very popular in image classification problems, leading up to region-based CNN's such as R-CNN [29], Fast R-CNN [30] and even Faster R-CNN [31]. Their goal was to solve the object detection problem: given an input image, the output should be a bounding box envolving the object and also a label assigned to the proposed region. In [29], the authors used Selective Search [20] to generate 2000 region proposals that had an higher probability of containing a meaningful object: the idea was to reduce the time spent looking at all of the image in a sliding window manner, like in traditional methods. After this stage, the region proposals were served as input to a trained CNN to extract features who would be given as input to a set of linear SVM's to perform classification. Fast R-CNN [30] noted that R-CNN took too much time in the training stage and also in the evaluation stage(up to 53 seconds per image) and developed an algorithm who was faster and with higher detection quality. Later on, Faster R-CNN [31] tried to enhance the complex training pipeline of its antecessors by inserting the region-proposal method after the convolutional layers, leading up to an even faster implementation. Recently, an improvement was made over Faster R-CNN: Mask R-CNN[] adds a small branch over Faster R-CNN in order to perform object segmentation on the predicted bounding box, working in parallel with the Faster R-CNN branch. Also, they found it crucial to change the Faster R-CNN part, replacing the RoIPool operation for RoIAlign to improve the mask accuracy by maintaining exact pixel spatial locations.

Later, the Inception Model appeared [2], a paper proposed by Google, who showed that CNN layers didn't need to be stacked up sequentially: some operations could be done in parallel. Also, they didn't use fully connected layers, replacing them with average pooling. In Figure 2.4 we can see an overview of the architecture.

Specifically on object detection, meaning that a bounding box over the object of interest is predicted and a label is assigned to it, YoloV3[32] is one of the most well-known approaches. Even though not as precise as, for example, RetinaNet[33], its purpose is to be able to run in real-time at a high frame rate (30FPS on a NVIDIA Titan X). A more thorough description of this algorithm will be presented in the Face Detection chapter.

2.3 Face Recognition

Face recognition is the process of identifying a person based on an image of its face. In other words, to be able to give a name to a face. Over the last two decades, face recognition has been improving a lot, reducing error rates by three orders of magnitude when recognizing frontal faces in constrained environments [25]. However, like in general object detection and recognition, the

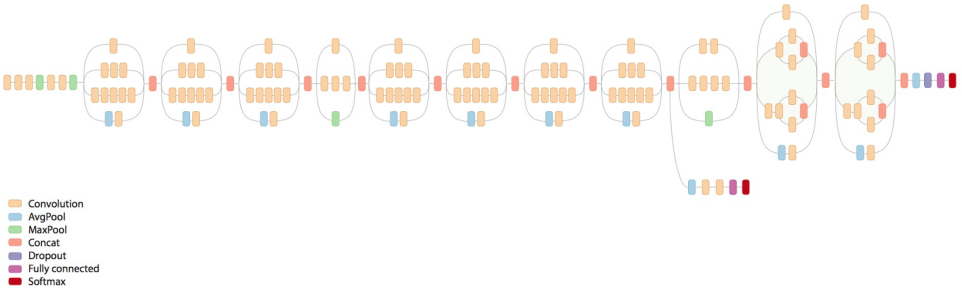

Figure 2.4: The Inception Model(GoogleNet) [2]. As we can see, some operations are executed in parallel.

most usual methods can't perform well in unconstrained environments, mainly because of their sensitivity to conditions like lighting, facial expression and occlusion.

2.3.1 Face Detection

Face detection was one of the most popular research topics in computer vision and pattern recognition, and it has been studied extensively over the past years. It is the first step to solve many face-related tasks, like face verification, face recognition and face clustering.

Like in usual object detection, older research methods focused more on extracting different types of known features and making a decision based on the designed feature vector, mainly through traditional classifying machine learning algorithms. These approaches are ineffective, in the sense that they often require the craft of effective features "by hand" and also because each of the extracted features has to be optimized separately [26].

In general, face detection can be considered as a particular type of object detection, because in computer vision terms, a face is an object. As such, in recent years deep learning methods have been employed to solve this task successfully, particularly by using deep convolutional neural networks, as seen above. Using these kinds of methods, the "hand-picked" features are avoided, and they are learned by the network, training it through an annotated database of images containing one or more instances of the object (in this case, faces).

2.3.1.1 Face Detection in the Visible Domain

Despite many studies before, it was not until the work of Viola and Jones [21] that the performance was satisfactory. Their work was the first to employ rectangular Haar features evaluated through an AdaBoost (adaptative boost) classifier, achieving real time face detection. However, it didn't cope well with non-frontal faces, and the feature size was relatively large [26]. Afterwards, other types of feature vectors were proposed and employed such as HOG [24], SIFT [22] and SURF [23], LBP and combinations of features. Using deep learning methods, [26] uses a methodology like in Faster R-CNN to detect faces, essentially consisting of a Region Proposal Network followed by

a fast R-CNN network to classify objects and at the same time refining the boundaries of those regions.

2.3.1.2 Face Detection in the Thermal Domain

In the thermal domain, [34] the well-known Otsu's method is used to convert the thermal image into binary form, followed by the calculation of the image's horizontal projection to determine the height and width of the head region. Similarly, [35] used region growing techniques and morphology operations followed by the derivatives of the horizontal and vertical projections to locate the minimal rectangle containing the face. In [36], the authors were able to enhance and apply the well-known Viola-Jones method to thermal images. They employed and tested several features, like LBP, HOG and Haar, and concluded that the performance of the system was better using LBP features. Also, they mention that the preprocessing of the thermal images using Otsu's method for binarization and an edge detection process can be an important step to reduce the false positive rate in face detection.

2.3.2 Face Recognition in the Visible Domain

Most current face verification methods employ hand-crafted features for detection, employing tens of thousands of descriptors for faces using Haar-like features, LPQ , LBP and Gabor-like features [37]. These types of algorithms achieved good performance in constrained face recognition, however it decreased considerably in unconstrained environments, like the images presented in LFW (Labeled Faces in the Wild) [38].

A ground-breaking result in face recognition tasks in the visible domain was achieved by DeepFace [25]. In their work, after the face detection step, the authors employ an algorithm to frontalize the face (alignment by a 3D model), followed by a nine deep neural network to classify the detected faces. They were able to achieve a 97.35% accuracy on LFW (Labeled Faces in the Wild) [38], a difficult and unconstrained dataset.

Another very well established work in this field is FaceNet [39], where the authors implement a very deep convolutional neural network trained using triplets constituted by two matching images and one non-matching face patch. In this training step, the goal was to minimize the distance between the two face patches while maximizing the distance to the non-matching face image. As the authors state, the main advantage of their method is the high representational efficiency, achieving an accuracy of 99.63% on the LFW benchmark while using only 128 bytes per face.

2.3.3 Face Recognition in the Thermal Domain

Using thermal images to solve the task of face recognition has some advantages over RGB images. Even though they are usually available in lower resolutions, they are more robust to varying conditions such as head pose, facial expression and lightning conditions [40]. Still, face recognition in the thermal domain doesn't receive as much attention as in the visible domain.

Various visible-domain feature-based methods have been implemented sucessfully in the thermal domain. [41] suggests using moments for features in face recognition. Moments can be defined as "scalar quantities used to characterize a function" [41], and according to the authors they are able to describe objects while being insensitive to deformations and providing enough distinctive attributes to sucessfully discriminate between classes. They achieved a rank-1 success rate of 97.5% tested on a database formed by images with different facial expression, light conditions and with different time-laps. Another feature-based method was designed by [40], where Gabor-like features were extracted from thermal images to produce a system robust to changes in scale and orientation. Afterwards, a Fast Independent Component Analysis (FastICA) was used to obtain the feature vectors over the Gabor-filtered images, followed by classification using linear kernel SVM's. They were able to achieve state of the art results, with a recognition rate of over 96%.

In an effort to try and find the best features for feature-based face recognition in unconstrained environments, [42] made a study comparing some commonly used approaches, namely: LBP, WLD(Weber Linear Descriptors), GJB(Gabor Jet Descriptors)[43], SIFT and SURF. They concluded that WLD-based methods were an excellent choice, either in terms of real-time operation (it was the second fastest algorithm, after LBP) and in recognition rate. Also, even though SIFT is not widely applied to face recognition tasks, it shows robustness regarding alignment errors, facial expressions and changes in rotation although lacking in terms of processing time. Even though nowadays CNNs are being widely applied in face recognition problems, we could only find one article [44] exploring these algorithms. In [44], the authors implemented a nine-layer convolutional neural network achieving state-of-the-art results like seen in [45] [46] [41]. The main issue continues to be the lack of big and labeled databases suitable for deep learning techniques in the thermal domain, even though deep neural networks could be of interest to researchers in the near future [18].

2.3.4 Face Recognition in the RGB-T Domain

By using or mixing the two modalities, we can harness the advantages of both: the high resolution and color information of the visible images and the temperature information, invariance to illumination, rotation and head pose made available by the thermal images. For this effect, various approaches have been developed, particularly three main methods: pixel-based fusion, feature-based fusion and decision-based fusion [47].

Pixel-based fusion methods suggest combining both the visible and the thermal image to explicitly form a new image. This implies a correct registration and alignment between both modalities. From then on, all the processing is done on that fused image, including feature extraction and classification. To achieve this, two weight factors should be used so that in the final image x% is composed by the thermal image and (1-x)% is composed by the visible image. The main issue regarding these kind of approaches is how to determine the optimal weights to apply in the process of image fusion. [47] proposes calculating two objective functions (one for each image domain) to evaluate both images, based on factors such as Renyi entropy(to assess constrast), sum of edge

intensities, and edgels. Then, Simmulated Annealing (SA) is applied to find the optimum weight factors, followed by LRC (linear regression classification) training and classification on the resulting images. They were able to achieve a maximum accuracy of 98.42% on the UGC-JU database using a weighting factor of 0.9452 for the thermal images, concluding that thermal images are more effective for face recognition than images in RGB-domain [47]. However, they state that using a small ammount of visible information enhances the results compared to only considering the thermal information.

Another proposition for data fusion was carried out by [48], suggesting image fusion in the discrete wavelet transform (DWT) domain. Decomposing both images up to the n^{th} level, the maximum intensity pixels at each 3x3 window were selected and kept, to preserve the most prevalent features. The enhanced image was later inversely transformed, producing a new fused image. Their optimized fusion method was able to achieve a recognition rate of 95.84% tested in the Equinox facial database. To solve the translation variance problem, [49] considered adopting the à-trous wavelet transform, training a Random Forest(RF) [50] to both select the best fusion weights considering entropy of the source images and also to perform classification in a later stage, achieving maximum accuracies of 99.07% for the UGC-JU face database and 100% for the IRIS benchmark face database.

Feature-based fusion methods suggest extracting features from both image domains and then combining them into an enhanced feature vector. This way, the best features from the visible image and the best features from the thermal image can be combined to attempt to improve the recognition rate compared to only making use of one type of features. For that matter, [51] proposed making use of descriptors based on LDP (Local Directional Patterns) and LBP for the visible spectrum and only LBP for the thermal spectrum. The final features were selected through genetic algorithm, designed to optimize the feature vector. They accomplished results better than the ones achieved using exclusively one modality. The best combination was LBP for both the image domains, reaching a recognition rate of 99.01%.

Decision-based methods suggest a parallel pipeline, performing independent feature extraction and classification to both image domains. In the end, the obtained results are combined and a final classification decision is made. In [52] the three different types of fusion schemes are compared, testing them in the Notre-Dame dataset, and for the decision-based method the authors employed a sparse approach. They concluded that the decision-based approach achieved the highest result.

2.4 Head Pose Estimation

In computer vision, the problem of head pose estimation can be seen as the process of determining the orientation of the human head, through analysis of digital images or video. Similar to other computer vision tasks, a good pose estimator should shows robustness to a series of usual image alterations, like occlusion, camera distortion, different light conditions, facial expression and the presence of visual artifacts like glasses and hats [53].

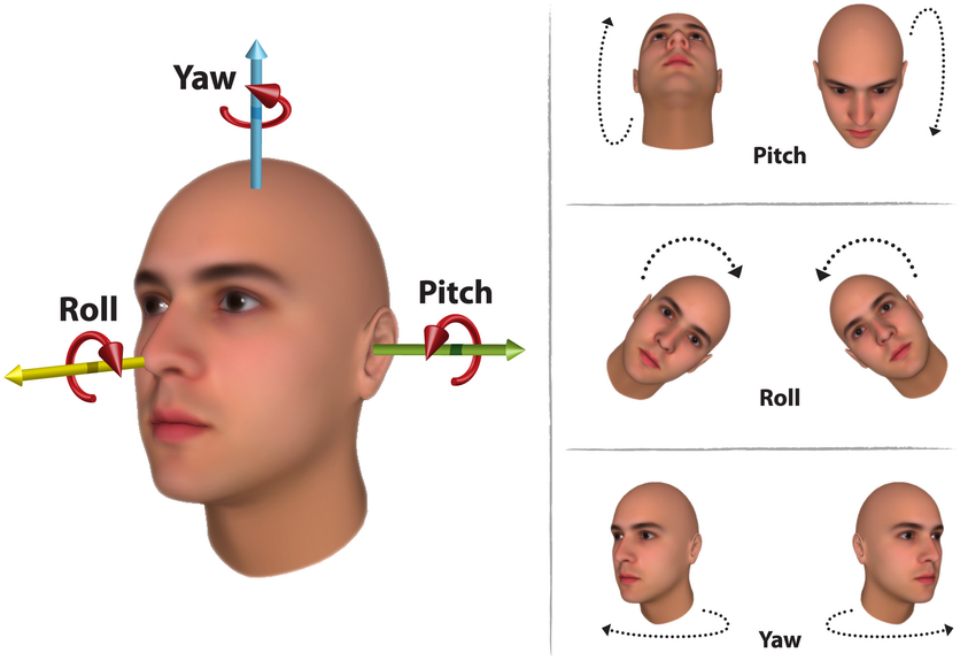

Figure 2.5: Roll, Pitch and Yaw. Taken from [3]

In this problem, an assumption is made that the 'human head can be modeled as a disembodied rigid object' [53]. As such, it can be defined as calculating the three degrees of freedom (DOF) of the head pose, namely roll, pitch and yaw, as shown in 2.5.

Several approaches normally applied to this subject will be discussed, namely: Manifold Embedding methods, Flexible Models and Regression methods.

2.4.1 Manifold embedding methods

Manifold Embedding methods attempt to reduce the dimensionality of data. In our case, an image can be seen as a sample in high-dimensional space, and this technique tries to down-sample the original data, representing it in a lower-dimensional space. Common techniques for dimensionality-reduction include PCA (principal component analysis) and LDA (linear discriminant analysis). According to [53], some other embedding approaches, like Isomap (Isometric feature mapping), LLE (Locally Linear Embedding) and Laplacian Eigenmaps (LE) have shown potential in the Head Pose Estimation problem. However, there is an inherent issue with this kind of approach: since it's an unsupervised learning algorithm, there is no way to guarantee that the achieved data representation varies directly with head pose.

In [54], the authors developed a driving database (Distracted Car Driver Database) to try and classify different pose angles in car drivers. After clustering the dataset into different classes using K-means Clustering, they applied PCA and LDA for dimensionality-reduction of the images.

Then, classification was performed using nearest neighbour algorithm between the training data and the input data.

2.4.2 Model fitting

Model fitting is probably the most used technique to solve the head pose estimation task. The first step of this approach is the detection of facial landmarks. Facial landmarks are specific points that can be found in faces that are used to label and identify key facial attributes in an image. The most common facial landmarks are the corners of the eyes, the tip of the nose and the center of the mouth, although other landmarks, like mouth corners and points along the jaw, can also be used. A very popular model used to detect facial landmarks is the one developed by [55], and it's available online. This algorithm is able to detect 68 points along the face in real-time and with high accuracy.

After the detection of facial landmarks, these points are then fitted in a 3D head model. Ideally, the 3D head model should match the head of the person, however a general 3D head model can be used [56]. Finally, we can use PnP algorithm or RANSAC [56] to iteratively calculate the head position and rotation. Even though this algorithm can achieve high performance, it has some issues in the facial landmark detection step (the points are probably not 100% accurate) and if a general anthropometric head model is used, instead of the person's head, the model will never exactly match the head of the subjects [57]. Regarding the number of facial landmarks to use, [57] suggests that there is a tradeoff: if we are certain that the extracted keypoints have enough accuracy, we want to use less of them. Otherwise, if we suspect that the measurements have some error we should use more points.

In the thermal domain, efforts are being made to employ this approach to head pose estimation. Due to the lack of annotated databases constituted by thermal images, [4] created a database with labeled points in the jaw, eyes, eyebrows, mouth and nose, with the purpose of training an active appearance model(AAM) [58]. Then, PCA was applied to allow the identification of the most significant parameters for the shape and texture of the faces, and model fitting was performed using a visible-domain alignment algorithm. They concluded that AAMs in the thermal domain were a viable approach. Example images can be seen in Figure 2.6. To enhance their results, the authors suggest a larger database, a customized fitting approach developed for thermal images, the usage of different landmarks (good RGB-based landmarks differ from good thermal-based landmarks) and better constrast-enhancing filtering techniques [4]. Meanwhile, an improvement on this work was released [59], achieving very good and robust results compared to their previous one.

2.4.3 Regression methods

Lately, non-linear regression methods are being sucessfully employed to the head pose estimation task. This approach involves estimating pose by learning a non-linear functional mapping,

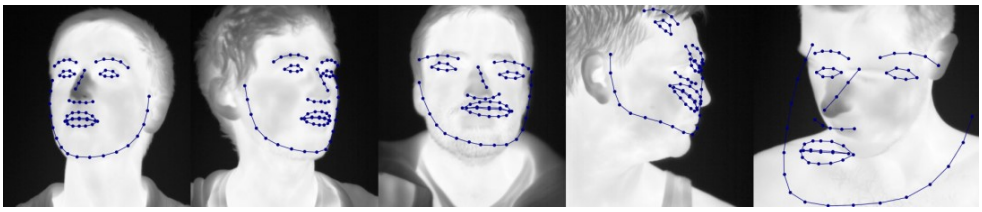

Figure 2.6: Results from [4]

to determine a pose direction from an image. Deep networks able to infer pose directly from images are a good solution for the head pose estimation problem, if a sufficient amount of labeled training data is available. The advantages of these kinds of solutions include robustness to high and low resolution images and also their 'representational ability in tolerating systematic errors in the training set data' [60]. Coupled with the recent developments in numerous computer visions datasets, these techniques can be key in mastering the head pose estimation task.

In [60], the authors make an evaluation of multiple CNN architectures and factors, comparing it with other algorithms, and prove that an approach based on CNNs, dropout and adaptive gradient methods can achieve state of the art results in human head pose estimation. They trained different CNNs, one for each degree of freedom, and with varying number of layers and parameters, to try and find the better model. The system was implemented in Python using Tensorflow and OpenCV, and it's released under an open-source licence, available for both academic and commercial purposes. Also related to deep learning techniques, [57] shows that implementing a multi-loss deep network can outperform landmark-to-pose algorithms. Using appropriate data augmentation techniques, their method shows robustness to very low resolution pictures and extreme poses.

2.5 Fatigue and Drowsiness Detection

In this section, we'll make a brief review on the state of the art on fatigue and drowsiness detection. Before that, it's important to make the distinction between fatigue and drowsiness, since the terms are related but slightly different: drowsiness refers specifically to the state right before sleep, while fatigue is related to the feeling of tiredness and exhaustion.

Systems that are able to detect if a person is tired or sleepy automatically are being very used in the present, specially in Driver Monitoring Systems (DMC's), as a way of preventing drivers from falling asleep on the wheel, a situation that is responsible for a lot of accidents on the road nowadays: according to the National Highway Traffic Safety Administration, "drowsy driving was responsible for 72,000 crashes, 44,000 injuries and 800 deaths in 2013". Criteria for fatigue evaluation may include yawning and blinking frequently, eye state classification (if it's closed or open), eye aperture and also frequent body position adjustment.

For fatigue and drowsiness evaluation systems, normally a RGB camera is used, although NIR cameras are also a good option. Mainly, two main options are employed: either estimating eye openness and closure or treating it as a binary classification problem (open or close).

2.5.1 Estimating eye openness

Using visible images, methods that rely on estimating the percentage of eye closure to determine if a eye is opened or not require using a facial feature point detector, so that the eye corners and the upper and lower positions of the eyelids are detected and localized. Then, the ratio of width to height of the eye can be calculated and a estimation about the eye state can be done. In [61], a total of five features obtained from the eye are used to estimate the PERCLOS measure, along with head pose estimation using POSIT algorithm, showing that combining eye and head information accomplishes a better classification accuracy. In [62], the authors estimate this measure by using a SVM to the edge image, calculated with the Sobel filter. In [63] a method for estimating the percentage of eye closure (PERCLOS) is developed using spectral regression, a powerful tool for dimensionality reduction and manifold learning.

A big drawback related to this method is that it requires a very precise localization of the eye landmarks, including corners and top and bottom pupils, which is very difficult to achieve.

2.5.2 Eye state classification

The classification approach turns the problem into a binary classification task: determining in each frame if the eye is closed or not.[64] uses the frame difference between two consecutive images and compare them using template matching, having one template for opened eye state and another for closed eye state, while [65] also calculates the frame difference image followed by determining its vertical projection. Image projection methods are also employed on the eye region[66], achieving fairly good results using a simple technique. [67] uses a weight matrix specifically made to detect the "strong line of dark values that correspond to eyelashes" that appears when the eye is closed.

Chapter 3

System Description and Dataset Creation

In this chapter, a brief description of the system is made, decomposing it into its parts and significant blocks, and we'll refer the tools used and all the dependencies used along the project. Also, the dataset will be described and its creation will be explained.

3.1 Problem Definition

The problem in hand can be defined by these different tasks: detecting how many people are inside the car (by detecting and counting the different faces inside the vehicle), recognizing who is inside the car at a given time (face recognition) and fitness monitoring of the passengers, specifically the estimation of their body temperature if a person is fatigued or tired (fatigue and drowsiness detection). Our goal is to design an algorithm capable of achieving these tasks using thermal images, and that is capable of running in an embedded platform with robustness and acceptable performance (as fast as possible) for every different conditions (lightning, temperature, ...) that we may encounter inside the vehicle anytime.

3.2 System Pipeline

In Figure 3.1, a block diagram corresponding to a top-level view of the system is presented.

As it can be seen in Figure 3.1, face detection is the first step in the pipeline, and every other application of the system is dependant from this block. Similarly, even though the face recognition subsystem doesn't need it, the facial landmark detector is a crucial step for the temperature monitor and the fatigue monitor, since these two objects depend on it to retrieve their inputs (eye corners for the temperature monitor and eye images (including the eyebrows) for the fatigue monitor).

It's also important to mention that, since face detection and facial landmark detection were two crucial steps for all the proposed final applications of the system and others (smoking detection and emotion recognition, for example), and since we were both doing this thesis with support from

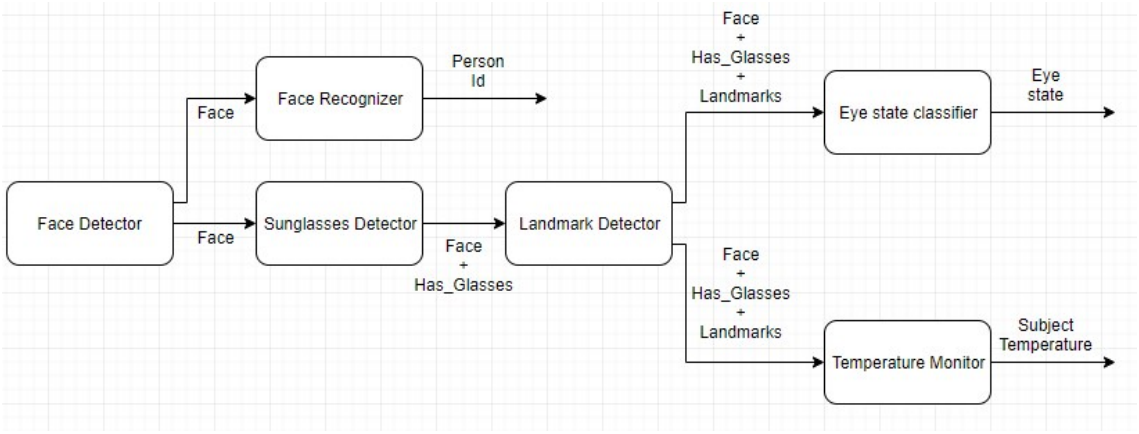

Figure 3.1: Block Diagram describing the system

System	***Input***	***Output***
Face Detector	Entire Thermal Image	Facial Bounding Box (x,y,width and height)
Glasses detector	Image cropped with Facial Bounding Box	Boolean representing glasses usage
Facial Landmark Detector	Image cropped with Facial Bounding Box + Glasses usage boolean	Location of facial landmarks (x,y)
Face Recognizer	Image cropped with Facial Bounding Box	Identity output (person ID)
Fatigue Monitor	Image cropped with Facial Bounding Box + Eye corners position + Eyebrow corners position	Eye state (open or close), Sleep/Microsleep detection
Temperature Monitor	Image cropped with Facial Bounding Box + Eye corners position	Body temperature estimation (numerical value)

Table 3.1: Inputs and Outputs of each subsystem

the same company (Bosch Car Multimedia S.A.), these two blocks were developed collaboratively with my colleague Gustavo Silva.

To make it clearer, Table 3.1 is presented, where the inputs and outputs of each system are explained.

3.3 Tools and Frameworks

All the code was written in python, except for the camera driver that was adapted from C. The frameworks used for developing our deep learning models were Keras and Tensorflow, and for image processing and display openCV was utilized. Every project dependency was installed through pip or Anaconda. The packages used are presented in Table 3.2

Packages	pip	anaconda
	bunch	cudnn=6.0
	tensorflow-gpu	cudatoolkit
	Keras	openCV
	Keras-vis	dlib
	imgaug	pytorch
	imutils	
	mtcnn	
	numpy	
	pandas	
	torchvision	
	tqdm	

Table 3.2: Packages installed for the thesis

3.4 Metrics Used

In this section, the metrics used along this dissertation will be detailed.

3.4.1 Classification accuracy

Classification accuracy is simply the number of correct predictions over the total number of predictions made. This metric is useful for classification problems. In this document, this metric is used to assess the precision on the face recognition task (top-1 accuracy), eye state classification and the sunglasses detector. The formula is presented in equation

$$Accuracy = \frac{\text{N(Correct Predictions)}}{\text{N(Total Predictions)}} \tag{3.1}$$

3.4.2 Average Precision

Average precision is the most common metric used to evaluate object detection algorithms. When the Intersection over union (IoU) of a bounding box prediction over the ground truth is bigger than a defined value, it counts as a correct prediction. As such, it is defined as the ratio between correct predictions and total predictions. Normal IoU threshold values are between 0.5 and 0.95, and mAP (mean Average Precision) is defined as the average of all the average precisions (AP's) in the range [0.5-0.95] with a step of 0.05 (AP(0.5), AP(0.55), AP(0.6), ..., AP(0.95)). The formulas for IoU and AP are presented in equation

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{3.2}$$

$$AP(x) = \frac{N(IoU >= x)}{N(IoU >= x) + N(IoU < x)} \tag{3.3}$$

$$mAP(x) = \frac{AP(0.5) + AP(0.55) + ... + AP(0.95)}{10} \tag{3.4}$$

Figure 3.2: Image taken from the RGB-D-T dataset. As we can see, there is no background

3.4.3 Mean Squared Error

To evaluate a procedure for estimating an unobserved quantity, a frequently used metric is the mean squared error (MSE), defined as the mean of the sum of the squares of the differences between the estimated value and the ground truth value. In this thesis, this metric is used to evaluate our facial keypoint detector. The equation to calculate this metric is presented below, where Y refers to pixel coordinates (x,y).

$$MSE = \frac{1}{N} * \sum_{i=1}^{n} *(Y_{\text{ground truth}} - Y_{\text{predicted}})^2 \tag{3.5}$$

3.5 Dataset Creation

3.5.1 Context

Due to a lack of labeled and complete datasets in the thermal domain, and since we were operating in a pre-defined environment (the inside of a vehicle), we felt the need to create our own dataset, specifically designed for our purposes.

After analyzing the publicly available datasets containing thermal images, we were only able to find two that were (kind of) correctly labeled: NVIE[68], containing images from asian people simulating(posed) and experiencing(spontaneous) different emotions such as happiness, sadness, surprise, fear, anger and disgust, and also the RGB-D-T[69] dataset, designed for face recognition, and containing facial images from 51 subjects in varying light, rotation and facial expressions. Another issue related to both these datasets is their simplicity: all the available frames only contain the subject and a simple background, with no added "noise", which means that using their images to train algorithms that are meant to be applied on situations where there will be different backgrounds and even other persons appearing in the footages may not be the best idea. Am example image from the RGB-D-T dataset can be seen in Figure 3.2

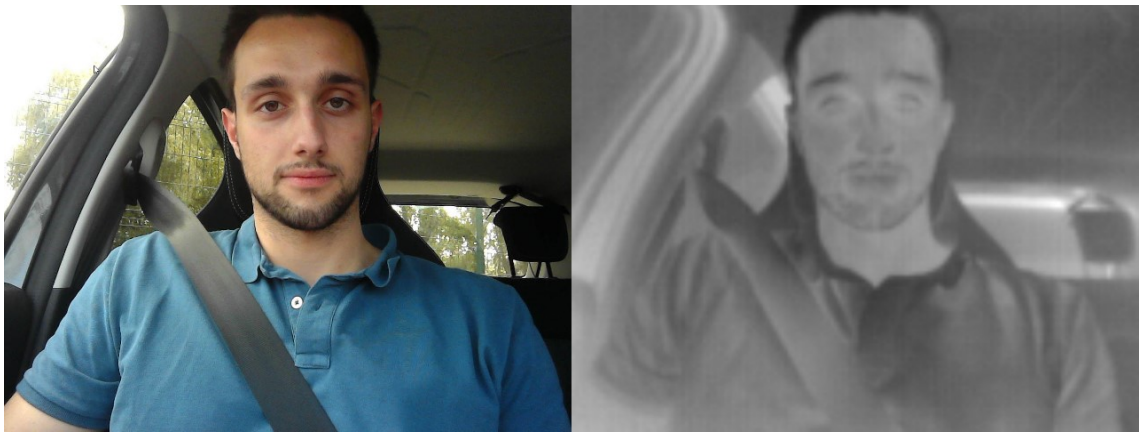

Figure 3.3: Example images taken from our dataset

Working with a big company such as Bosch, and even though we didn't had a lot of time, we had the opportunity to create a big and robust dataset, with lots of people(workers from the company), suitable to our planned objectives. To ensure proper video recording and to guarantee that no time was being wasted and nothing would go wrong or forgotten during the recording, we developed a script containing in detail the actions that the test subjects should perform in every instant, such as head roll, pitch and yaw movements for proper head pose estimation and landmark detection, different facial expressions to simulate emotions, and "fatigue simulation movements" such as change in eye blinking rate, yawns and closed eye time. Only one subject was recorded at a time, seated in the passenger seat. The mentioned script can be seen with full detail in Annex 1, and images from the dataset can be seen in Figure 3.3

Due to the new data privacy law [70], which entered in motion on the 25th of May of 2018, face images from people and even name and gender of the subjects should not be kept unless explicitly allowed. To account for this new law, a consignment letter had to be written with relevant information regarding the script and the context and purpose of the images, so that the person would know exactly the intention behind the recordings. For us to be able to capture the images, the person should have signed this document beforehand.

3.5.2 Camera Setup

As specified in the previous chapter, the dataset was recorded by taking pictures of the subjects individually. As such, we decided to place the camera in front of the person, using a standard mount fixed on the frontal glass of the car and holding our camera on the passenger dashboard.

Since one of the purposes of the dataset was to check for the possibility of measuring a person's respiration rate using a thermal camera (not tackled in this dissertation, but can be done in the future), by detecting the temperature differences in the nostrils caused by the subject's breathing pattern, the only requirement for the camera position (with the exception of the obvious one,

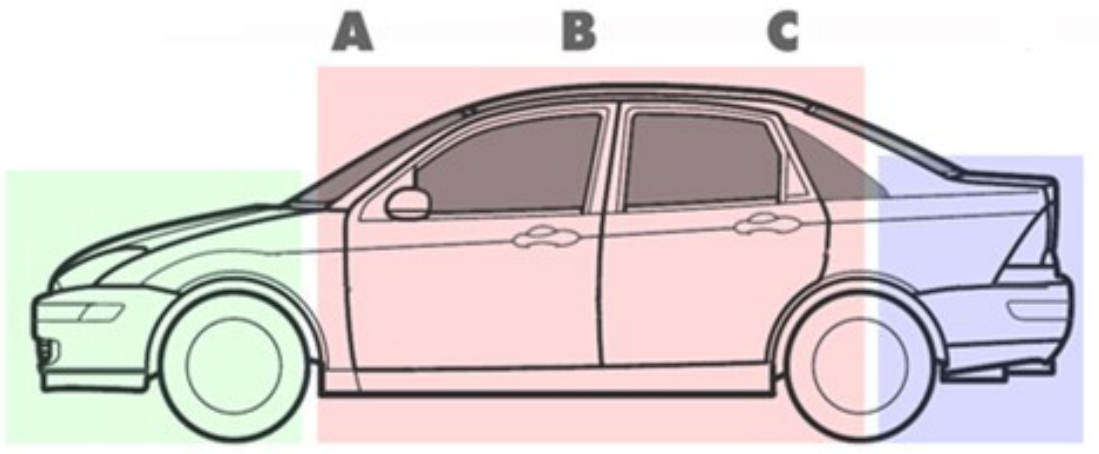

Figure 3.4: Camera setup view from outside the car. Images were collected from point A

getting the subject's face on the frames) was that the camera was placed slightly below the individual's face, so that a partial bottom-view of the person's nostrils was captured by the thermal camera. Pictures of the acquisition setup can be seen in Figures 3.4 and 3.5

3.5.3 Camera used

The camera used was the FLIR ONE Pro. It was chosen due to its low price and because it includes the two modalities (RGB and Thermal) that we were researching. Briefly, it offers a thermal sensor that captures images with 160x120 resolution at 8.7Hz and in the 8-14 μM spectral range, with Horizontal field-of-view (FOV) of 55^{o} and Vertical FOV of 43^{o}. Another key advantage is its automatic thermal calibration, using its shutter. The camera is designed to be used with the cellphone, available with Male USB-C for Android Smartphone or Male Lightning for iOS.

Using the cellphone app, we weren't able to extract the RGB and the pure thermal image from the camera. Because of this, there was a need to develop a driver, so that we could control the camera and save the images directly to the computer and in the desired folder form, and additionally that allowed us to extract both the RGB and the pure thermal image, with the camera's specified thermal resolution (14bit). By pure thermal image, we refer to the matrix containing the raw temperature values captured by the camera. It was important for us to get the raw temperature values, for temperature-sensitive applications such as temperature monitoring and respiratory rate calculation, and also to be able to estimate the indoor temperature inside of the vehicle.

With this in mind, we searched the web for a solution, and ended up downloading an open-source driver from Github https://github.com/fnoop/flirone-v4l2 to operate with the FLIR One Pro camera on Linux, based on Video4Linux (v4l2). After installing its requirements and altering some details, we performed a recording test, where we found out that the two frames (RGB and Thermal) extracted from the camera were not synchronized and sometimes were repeated. To narrow down the problem, we analyzed the driver and concluded that one possible

Figure 3.5: Camera setup view from inside the car.

fix was to remove v4l2 from the pipeline and simply incorporating the image writing step directly inside the driver program: since the driver was originally written and compiled in C, and because we were using openCV to save and display the images (available in Python and C++), we opted to re-compile the driver with a C++ compiler (gcc) and linking the openCV library. To record the images inside the car, a Jetson TX2 was used, controlled from a cellphone through Secure Shell(SSH) protocol.

3.5.4 Folder structure and annotations

Since we couldn't keep personal names, due to data privacy laws, people were labeled based on numbers, by order of recording (later on they were scrambled). Each folder was identified by this number, so that no other identification was needed, and each folder contained two subfolders, named RGB and Thermal, containing images from both types.

Each folder also had an info.csv file, containing all the labeled frames from a subject and also some hand-labeled frames with relevant information (time and number of frame, if the subject was wearing glasses or smoking his eye state and facial expression). Lastly, each person's folder contained an annotations.xml file, where facial bounding boxes and landmarks were saved.

The visible image was saved as 8bit JPEG format with the resolution provided by the camera, while the thermal frame was saved as 16bit PNG (to take advantage of the 14bit dynamic range of the thermal sensor) with pixels representing the temperature absolute value: this means that, since the camera provides a large temperature range(-20^{o}C to 400^{o}C), and because ambient temperature and human body temperature fall in the low-end of that spectrum, the images without any type of

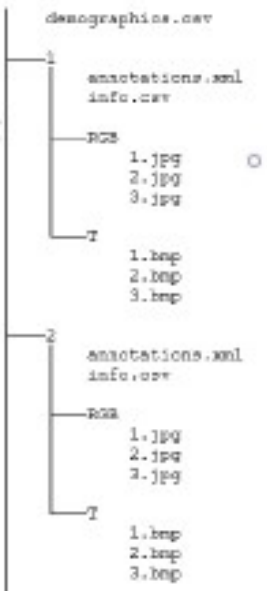

Figure 3.6: Directory tree of the Bosch (RGB+T) dataset. Each folder contained a person's annotations, labeled frames (info.csv) and the corresponding RGB and T images

Number of Subjects	***Average Age***	***Average Height***	***Ethnicity***
38	28,84 +- 10,11	1.76 +- 0.10	All white
Beard	***Moustache***	***Hair Size***	***Gender***
24 yes, 14 no	25 yes, 13 no	3 bald, 5 long, 3 medium and 27 small	34 male, 4 female

Table 3.3: Dataset Demographics

temperature equalization would appear to be basically black. The directory hierarchy can be seen in Figure 3.6

3.5.5 Demographics

Some information was collected about the subjects, so that we could evaluate our algorithms regarding natural differences in the recorded persons, such as height, age and hair size. In Table 3.3 relevant information is presented about the dataset.

3.5.6 Image Labelling

Using the FLIR One PRO, and because this camera possesses an RGB and a thermal camera side by side, we had the opportunity to save a lot of time by developing a method to automatically label the images, by using highly accurate pre-trained RGB algorithms to label the thermal images. So, after searching the web for the best RGB face detectors and facial landmarks detectors, we settled with MTCNN[71], used by FaceNet[39] for face detection, downloaded from `https://github.com/ipazc/mtcnn` and also hourglass CNN[72], downloaded from

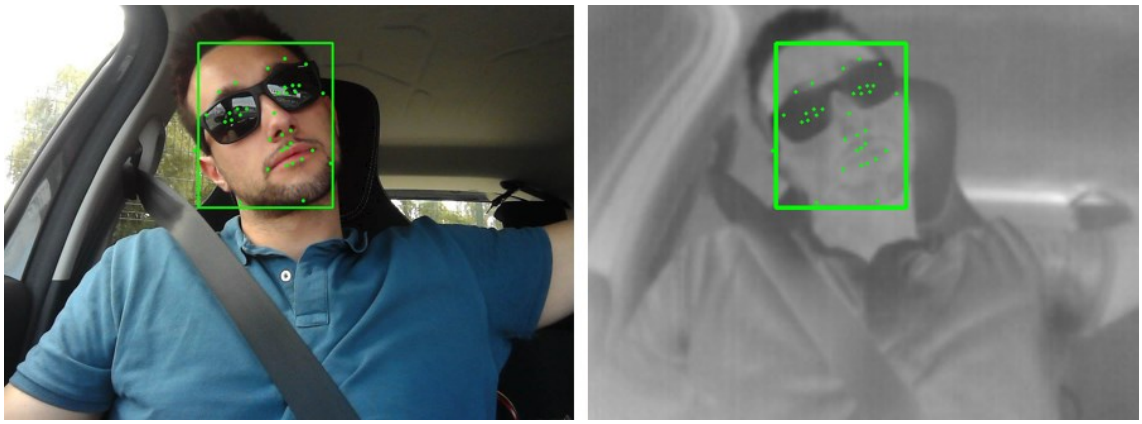

Figure 3.7: Labelled images from our dataset

https://github.com/1adrianb/2D-and-3D-face-alignment, for the localization of facial landmarks. Running eye tests with these algorithms, we concluded that they were the best amongst the ones tested, and indeed they are very accurate.

To be able to automatically label the thermal images with the RGB images, both frames had to be correctly aligned. To do this, we overlayed them both (0.5*RGB + 0.5*Thermal) and simply cut the borders of the RGB images, so that both frames would match each other. Since this alignment was dependant on the distance from the object to the camera, and even though in a final solution the camera should be placed always in the same spot, there was still some natural variation on camera position (between each recording session) and even seat position, affecting the correct alignment of both frames. To account for this, a semi-automatic labelling process was employed: after running a script to automatically save the RGB (and equivalent thermal) facial bounding box and facial landmarks, we would then check the images with the annotated points displayed: if we thought it was well done, we would save that annotation; if not, we would discard it. With this procedure, we were able to filter out the bad labels performed by the RGB algorithms (naturally, they were not 100% accurate) and also filter cases where the alignment was not perfect.

Per subject, one recording would amount for around 3200 frames. With the frame rate of around 9 frames per second (fps), we noticed that consecutive frames were very similar to each other, so to save us some work and time and to get more distinguishable frames, we chose to label the frames with a step of 10: if, for example, we started at frame 7, we would then label frames 17,27,37, and so on. In the end, we would end up with about 300 frames per subject. If, for some reason, we felt that we needed more images per person, we could simply re-do the labelling process with a smaller step. In figure 3.7 we can see an example labelled image from our dataset.

3.6 Conclusions

In this chapter, a general view of the whole system was shown, and the creation of the dataset was explained. As mentioned before, it was very important for us to have our own dataset due to the

lack of good datasets available for commercial purposes in the thermal modality, and also because it allowed us to develop the algorithms much more efficiently, since it was made specifically just like we wanted and for our purposes).

Chapter 4

Proposed Solution

In this chapter, our final solutions for the proposed objectives will be explained, followed by their respective results and conclusions.

4.1 Face Detector

Face detection has been studied extensively over the recent years, mainly because it is a crucial step to allow for other more complicated tasks such as face recognition. In the thermal domain, it is common to see approaches that are more traditional such as Otsu's threshold[34], horizontal and vertical projections[35] and also the largely used Viola-Jones[36] for the exercise of estimating facial position.

4.1.1 Approaches

In the beginning of this thesis, and after analysing all the related work concerning this topic, we decided to try and replicate their papers, namely calculating horizontal and vertical projections, hand-designed feature extraction (Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Gabor) followed by classification using Support Vector Machines (SVM's)[73], Otsu's segmentation[34] and also the very popular Viola-Jones method[36]. After their implementation, and even though we currently do not have numerical results to show it, the general conclusion was that these types of algorithms can only provide good results in (very) constrained environments, like fixed frontal face positions, regular backgrounds and controlled temperatures, and when the subject adopted a different head pose or the scenario changed the results were much worse.

4.1.2 Solution

Based on the recent trend of using machine learning algorithms for object detection, we decided to try an approach like this. After analyzing the most common algorithms, and with our requirements (high accuracy and capable of running in real time), we decided to use YOLO[32], resorting to

(2).png (2).png

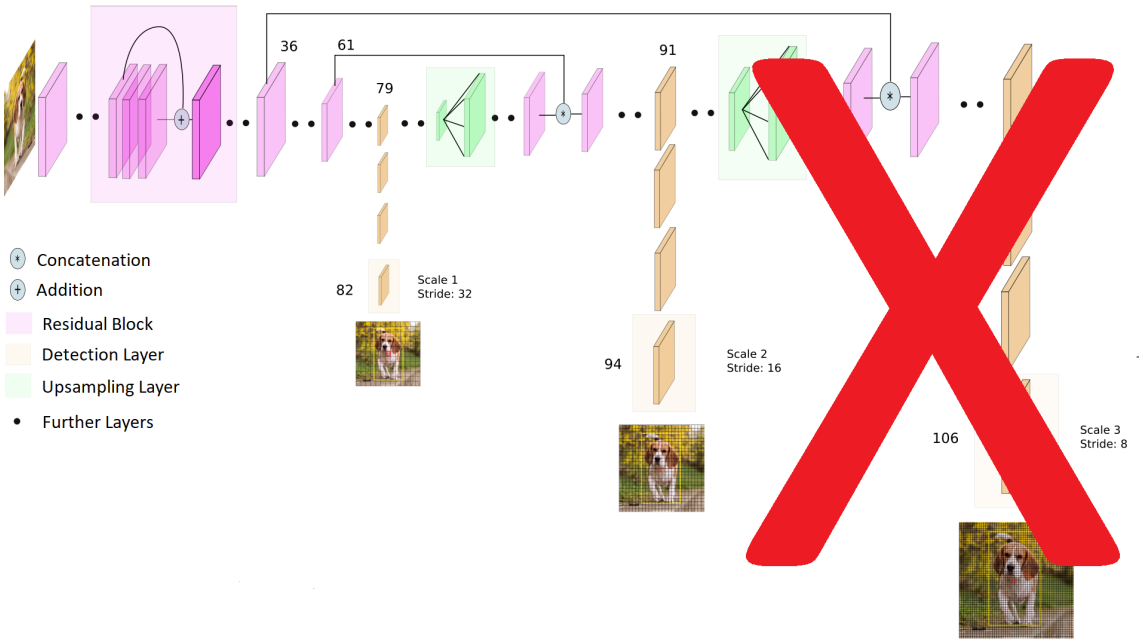

Figure 4.1: YoloV3, adapted from [5] We can see the removed part of the network marked with a red X.

transfer learning and training over weights pre-trained with ImageNet[74] to detect only the class that was of use to us, faces in thermal images.

On the start date of this thesis, only YOLOv2 was available, but during its development YOLOv3 was released. When the newer version came out, we had already implemented the previous version, so we ended up updating it and re-training it with the newer images that were recorded in the meantime.

4.1.2.1 Technical Review

YOLO[32] stands for You Only Look Once, and it's a state-of-the-art real time object detection algorithm. As a benchmark, it can process images at 30 frames per second (FPS) on a Pascal Titan X and is able to achieve 57.9% mAP on COCO test-dev. As the name suggests, YOLO is an algorithm more focused on real-time performance, achieving worse results than for example [31][33] but with a much faster inference time.

According to its author, the YOLO model has several benefits over classifier-based systems: it looks at the whole image to make its prediction, meaning that the inference is done with the input's global context in mind. Also, because it looks at the whole image only once, it is capable of making its predictions several times faster.

In YOLOv2, the authors used a feature extractor network composed of 19 convolutional layers and 5 maxpooling layers, calling it Darknet-19. In YOLOv3, they decided to use a deeper

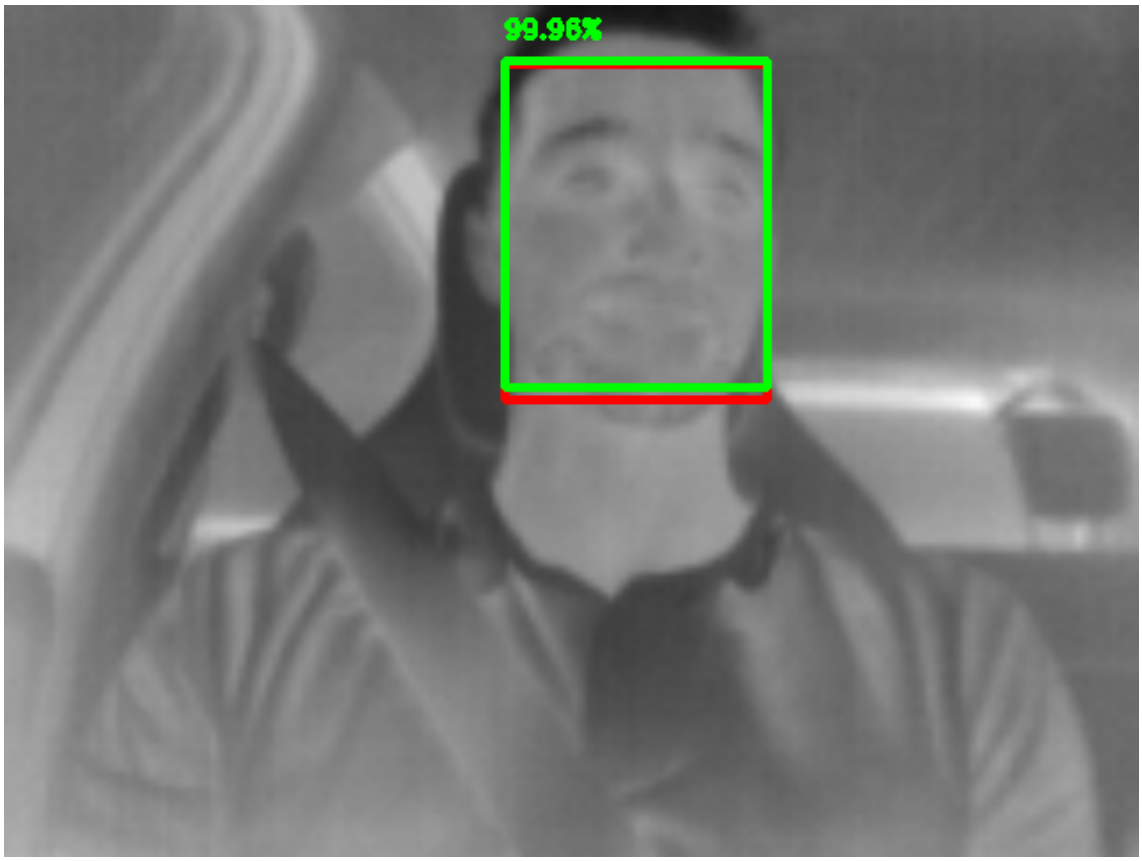

Figure 4.2: Example images. In green, the bounding box prediction from our system, and in red the ground truth. On top of the boxes the predictor's confidence is shown.

network, by adding residual layers, skip connections and replacing the downsampling procedure (maxpooling in YoloV2) by convolutional layers with stride 2, effectively reducing the previous feature map by a factor of 2. As stated in their paper, this new network is much more powerful than Darknet-19 even though it runs slower, and still more efficient than networks such as ResNet-101 and ResNet-152.

One of the problems with YOLOv2 was that it struggled with smaller objects, due to the loss in fine-grained features caused by downsampling the input in some layers. YOLO v3 performs detections at three different scales, by performing predictions on feature maps at three different places in the network. This procedure allows the preservation of fine grained features that are more relevant for small object detection. As seen in Figure4.1, this procedure is done at layers 79, 91 and 106.

Just like YOLOv2, the third version of YOLO uses the concept of anchor boxes. Basically, after some clustering studies on general ground truth labels, it seems that most bounding boxes have similar aspect ratios, so instead of directly prediction a bounding box, the YOLO algorithm predicts off-sets from a pre-defined set of boxes with pre-defined height-to-width ratios, called anchor boxes.

4.1.2.2 Image pre-processing and data augmentation

As input to the mentioned algorithm, grayscale images were used, equalized from 25°C to 43°C. To enhance our dataset, we used data augmentation. This way, we could generate images that were harder for the model to predict and therefore training it for more difficult cases. The augmentations used per image (applied to 50% of them) were: horizontal flip, image rotation, image translation, image blur (one of gaussian, average or median per image) and some brightness and contrast modifications.

To apply those augmentations to our images, we downloaded a python package named imgaug that provided easy to use functions for the task, including the needed adjustments in bounding boxes caused by the displacement and alteration of the original images.

4.1.2.3 Transfer learning with YOLO

We used transfer learning to train our single-class YOLO detector, meaning that we took advantage of the already trained weights available. The original network is designed to work on visible (RGB) images and to predict multiple classes. In our case, we only needed to localize and detect one class (faces) and our input images had only one channel (temperature).

Because YOLO works with anchor boxes, we used the K-means algorithm to cluster the aspect ratio of all the facial bounding boxes in order to generate anchor boxes designed specifically for our use case: this would result in better bounding box predictions and consequently an increase in mAP.

As described in the Technical Review section, YOLO makes predictions on three different points of the architecture, each one better suited to predict objects of a specific size. Since the only class to predict was faces, we didn't need the output of the small objects predictor, so we removed those last layers of the network. This resulted in a faster inference time, from 69ms to 48.3ms. The removed layers can be seen in Figure 4.1

Another issue that we had to deal with was the input channels: we only had one channel, and YOLO expected three channels. To account for this, several approaches were tried to design the best model: applying our camera's color palette to the temperature image (resulting in a 3-channel image), converting the grayscale image to an RGB image (replicating accross all three channels) or adapting YOLO to work with one channel as input, discarding the kernels of the first convolutional layer and using random weights, or by summing the kernels from the three channels. To determine the best model, we used the AP50, AP75 and the mAP metrics.

4.1.3 Results

For each option, the model was trained five times, and their average values were calculated. The results from these experiments described above can be seen in Table 4.1. This test was performed at a time where our dataset was comprised of only 18 people, so these results were obtained by training with our dataset subjects from id=1 to id=11 and validated using the subjects from id=12 to id=18.

Option	*AP25*	*AP50*	*AP75*	*mAP*	*time (s)*
3 Channels, Palette	96.10%	96.10%	70.49%	59.24%	0.0594
3 Channels, Grayscale	95.53%	95.25%	83.24%	64.41%	0.0576
1 Channel, sum of 3 channel weights	97.02%	96.98%	83.77%	65.85%	0.0483
1 Channel, random weights 1st layer	96.06%	95.86%	81.51%	65.03%	0.0544

Table 4.1: YOLO model tests

Test number	*AP25*	*AP50*	*AP75*	*mAP*	*time*
Test #1	99,61%	99,61%	97,17%	79,12%	0,044846
Test #2	99,48%	99,48%	97,00%	79,44%	0,054049
Test #3	99,32%	99,32%	96,63%	78,31%	0,068841
Test #4	**99,60%**	**99,60%**	**98,06%**	**79,47%**	**0,085061**
Test #5	99,54%	99,54%	95,15%	77,11%	0,101049

Table 4.2: Best model tests

As it can be seen on Table 4.1, the best model was achieved by using the sum of the kernels from the three channels, reaching a very satisfactory mAP value of 65.85% and being the fastest regarding inference time.

When we closed the dataset, with a total of 38 persons, the same model was re-trained using every subject. The model was trained five times and the best one was saved. Subjects from id=1 to id=30 were used for training and the subjects from id=31 to id=38 were used for testing. The results are shown in Table 4.2

The model that was kept was achieved in test 4, based on the mAP metric (79.47%).

4.1.4 Conclusions

An accurate and robust face detector using thermal images was developed, capable of running in real time, that can be used for applications such as counting the number of people inside a vehicle and as a first step in a pipeline for facial landmark prediction and face recognition. The final model achieved very satisfactory results in terms of AP.

4.2 Glasses Detector

In thermal images, and because regular glass does not transmit above wavelengths of around 3 µm, when a person is wearing any kind of glasses (normal or sunglasses) they appear as opaque. Since we have applications that are dependant of detecting eye landmarks, we had the need of developing a classificator to determine if a person is wearing glasses or not, so that we can decide if our predictions in those systems (temperature monitoring and eye state classification) are valid or not. For this effect, we developed a convolutional neural network capable of determining if a person is wearing glasses.

Type	***Filters***	***Size***	***Output***
Input Image	-	-	(30,60,1)
Convolutional	32	3x3	(28,58,32)
Batch Normalization	-	-	(28,58,32)
Activation (ReLU)	-	-	(28,58,32)
MaxPooling	-	2x2	(14,29,32)
Convolutional	32	3x3	(12,27,32)
Batch Normalization	-	-	(12,27,32)
Activation (ReLU)	-	-	(12,27,32)
MaxPooling	-	2x2	(6,13,32)
Fully-Connected	-	-	(32,)
Activation (ReLU)	-	-	(32)
Dropout (50%)	-	-	(32,)
Fully-Connected	-	-	(1,)

Table 4.3: Glasses-detecting CNN

	Training Accuracy	**Validation Accuracy**	***Batch Size***	**Loss Function**	**Optimizer and Learning Rate**	**Epochs**
Final model	98.26%	98.02%	8	Binary Cross-Entropy	Adam(lr=0.001)	6

Table 4.4: Information about final model

4.2.1 Proposed Solution

For this application, a convolutional neural network was developed. An important requirement of this subsystem was that it only depended on our face detector, so that a decision regarding glasses usage was made before the facial keypoint detector.

4.2.1.1 Image pre-processing and data augmentation

As input to this network, images cropped from facial bounding boxes starting from the top 10% until 50% were used, thermally equalized from 33°C to 45°C and resized to (60,30). Standard data augmentation techniques employed along this thesis were used, like image rotations and translations, blurs and contrast modifications.

4.2.1.2 Network architecture

As shown in Figure 4.3, glasses appear as black on the input images. With this in mind, and because we were dealing with a simple problem, only two convolutional blocks were used. Batch Normalization was applied after each convolutional layer and dropout was added after the fully-connected layer to reduce overfitting.

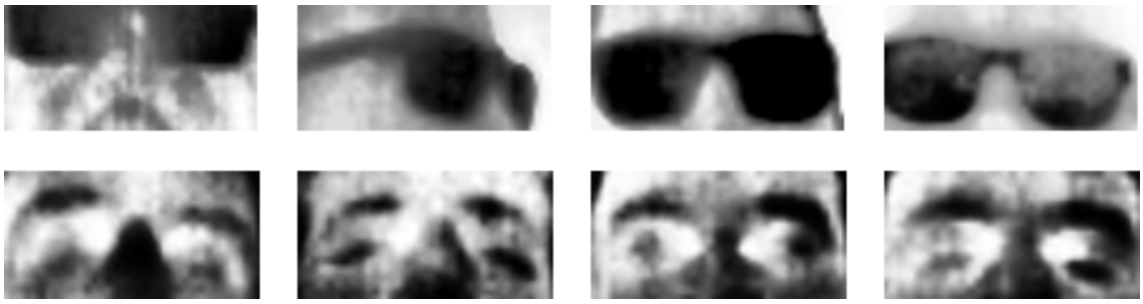

Figure 4.3: Input images to the glasses detector. It can be seen that glasses appear as black and it's easy to tell if a person is wearing glasses or not.

4.2.2 Results

The network was tested on the test set, and achieved 97.96% accuracy.

4.2.3 Conclusions

An effective model was developed to predict if a person is wearing glasses using thermal images. This was an important step for our temperature monitoring system and also the fatigue monitoring system. We are happy with the achieved results.

4.3 Facial Landmark Detector

4.3.1 Approaches

As in the rest of the chapters in this thesis, the first approach taken was to attempt to replicate the general state of the art, which was not very rich. So, we first started by using classic computer vision methods like Harris corner point detection[75] and hotspot detection for the eye corners. Once again, these variety of techniques are not very robust and work only in very constrained environments.

After exploring the state of the art in the visible domain, we discovered some interesting algorithms, namely: "Face Alignment at 3000 FPS via Regressing Local Binary Features"[76] and "One Millisecond Face Alignment with an Ensemble of Regression Trees"[55].

During the development of this thesis, a paper came out[77], where the authors achieved good results in facial keypoint detection, synthesizing RGB images from thermal images using edge-based features, and then applying the pre-trained model available in dlib[78].

4.3.2 Proposed Solution

Since there were virtually no solutions implemented for face keypoint detection in thermal images at the start of this thesis, we decided to try and implement the common algorithms used in the visible domain, particularly an ensemble of regression trees, inspired by [55].

4.3.2.1 Technical Review

Decision tree learning is an approach frequently used in data analysis, where the goal is to design an algorithm that estimates the value of an output variable based on several input variables. In each node, a decision is made relative to one (or more) input variables. A tree can be learned by splitting the training set into subsets, and this process is repeated in each subset in a recursive manner. In our case, the input variables are the images (pixel values) and the output is a given pixel localization. The advantage of regression trees is that they are able to explore and highlight complex or hidden relationships in the data, and for this reason they are used for getting well-fitted models that represent very accurately the data behaviour and which are particularly useful for prediction[79].

We are using an ensemble of regression trees, a predictive model composed of a weighted combination of multiple regression trees. Using multiple regression trees should increase predictive performance. There are two types of ensemble methods, boosted trees or bagged trees: the first one, used in AdaBoost and Viola-Jones, incrementally builds an ensemble by training each new instance to emphasize the previously trained ones, while the second one builds multiple decision trees by repeatedly resampling training data and voting the trees for a consensus prediction.

In our case, we are using a Random Forest, a type of bagged decision tree that tries to correct for the habit of decision trees to overfit the training set. In general, trees that are too deep tend to overfit the training set (low bias, high variance), and Random Forests are a way of averaging multiple deep decision trees trained on different parts of the same training set, with the goal of reducing the variance. This means that while the predictions of a single tree are highly sensitive to noise in its (individual) training set, the average of many trees is not, as long as the trees are not correlated. Training many trees on a single training set would give strongly correlated trees (or even the same tree), so these kinds of methods are a way of de-correlating the trees by showing them different subsets extracted from the same training set. Once the best regression tree has been obtained, it's necessary to prune it to avoid overfitting. This can be done with cross-validation, a technique widely used for selecting predictive models where several training instances are made using different subsets for training and validation.

4.3.2.2 Implementation

To train the algorithm, we used the dlib framework which had a function available to train an arbitrary shape predictor with custom images. This was much easier than implementing the training algorithm from scratch, although to use it, we needed to have all the images that we wanted to include in the training set inside a folder, with a file named "annotations.xml" containing all the annotations from all the images inside that folder. Accordingly, our solution involved iterating through all our training images, pre-processing them, saving them in a folder and converting the annotations from our format to the "dlib format". In the end of the training session, we could delete the folder.

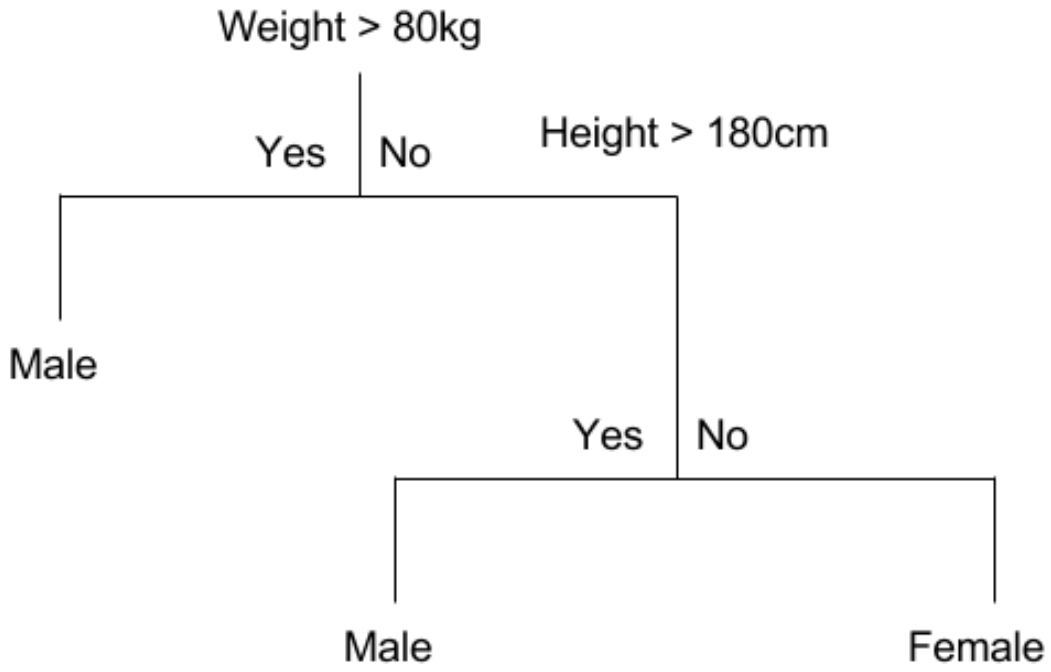

Figure 4.4: An example of a simple regression tree. In each node, a decision is made regarding an input variable

When a person is wearing glasses, it is impossible to see and predict eye landmarks with thermal images. As such, it doesn't make sense to try and guess where those landmarks are if we can't seem them. Because of that, using our glasses detector described in section 4.2, we developed two predictors: one for persons wearing glasses, that predicts 17 points, and one for persons not wearing glasses, that predicts 35 points. The points excluded from the "with glasses" predictor are the ones envolving the eyes and the eyebrows.

4.3.2.3 Image pre-processing and data augmentation

As input to the mentioned algorithm, grayscale images were used, and histogram equalization was applied to enhance their contrast. To amplify our dataset, we used data augmentation. This way, we could generate images that were harder for the model to predict and therefore training it for more difficult cases. The augmentations used per image were: horizontal flip, image rotation, image translation, image blur (one of gaussian, average or median) and some brightness and contrast modifications.

To apply those augmentations to our images, once again we used imgaug that provided easy to use functions for the task, including the needed adjustments in facial keypoints and bounding

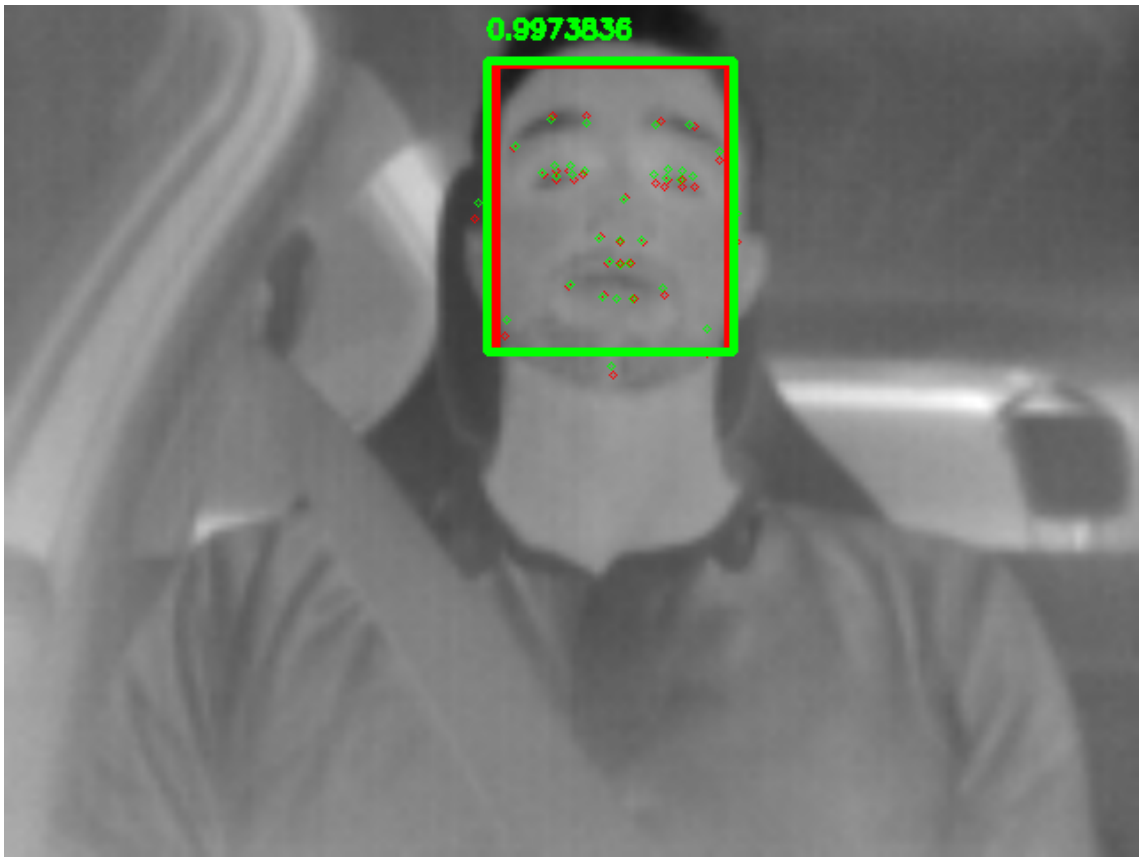

Figure 4.5: Facial landmark predictions: in green our predictions, and in red the ground truth keypoints

boxes caused by the displacement and alteration of the original images.

4.3.2.4 Parameter Optimization

To train the ensemble, a couple of parameters can be specified, namely: nu, the regularization parameter in the range (0,1], and tree depth, the depth of the trees used in each cascade. To find the optimal parameters and the best model, several experiments were performed with different parameter values, described in Table 4.5

After calculating the optimal parameters, tests were made using subjects that were not included previously. Two thermal predictors were developed: one previewing both cases (glasses and no glasses) and one general (same predictor for both cases).

4.3.3 Results

The results from the experiences described above can be seen in Table 4.5. The visible mean squared error (MSE) comes from predicting the equivalent RGB image using the pre-trained landmark detector available with dlib. The model was trained and tested always with the same subject: we are aware this may not be the best approach, but the script took too long to run for us to perform

Parameters tested	*Visible MSE*	*Thermal MSE*
nu = 0.002, depth = 4	8.56028E-5	3.6817E-4
nu = 0.002, depth = 5	8.56028E-5	3.4981E-4
nu = 0.002, depth = 6	8.56028E-5	3.5689E-4
nu = 0.002, depth = 7	8.56028E-5	3.5873E-4
nu = 0.003, depth = 4	8.56028E-5	3.7778E-4
nu = 0.003, depth = 5	8.56028E-5	3.5880E-4
nu = 0.003, depth = 6	8.56028E-5	3.9754E-4
nu = 0.003, depth = 7	8.56028E-5	3.7849E-4
nu = 0.004, depth = 4	8.56028E-5	3.9847E-4
nu = 0.004, depth = 5	8.56028E-5	4.1718E-4
nu = 0.004, depth = 6	8.56028E-5	4.2572E-4
nu = 0.004, depth = 7	8.56028E-5	3.8943E-4

Table 4.5: Facial keypoint detection model optimization

cross-validation. As we can see from the results table, all of the thermal MSE values are similar to each other, and for that test set the best model achieved was using nu = 0.002 and tree depth = 5.

The final test results are seen in Table 4.6, using ground-truth bounding boxes and using our face detector predictions. As shown, the facial landmark works better with our face detector than with the ground truth ones, reminding that the ground truths are not perfect since they were labeled automatically and further re-aligned with the thermal images. Also, it can be seen that our developed predictor outperforms the pre-trained one available online with dlib for the same images (the rgb predictor was evaluated on the corresponding RGB image). Also, we conclude that training two predictors to predict cases where the test subject is wearing glasses or not is not better than one trained without that distinction.

4.3.4 Conclusions

In this chapter the approach that we took was described, where we developed an usable and accurate facial keypoint detector made for thermal images. As future work, other approaches could be employed (CNN's), and other applications can be derived from this keypoint detector, such as facial alignment and head pose estimation.

Bounding Box	Thermal MSE Two predictors	Thermal MSE Single Predictor	Visible MSE
Ground Truth	6.4995	2.673	4.2095
YOLO prediction	5.8985	2.788	4.5850

Table 4.6: Landmark predictor final results

4.4 Temperature Monitoring

Face temperature monitoring using thermal cameras is used for a long time in airports, mainly for the widespread prevention of infectious diseases (manifested by fever) such as EBOLA, SARS and Tuberculosis[10] . Body temperature can be estimated by measuring the temperature in the inner corner of the eyes. Several studies show the correlation between the eye corner and the axilla (the commonly used point for body temperature measurement)[12][13]. Eye corners are also usually the hottest point in the face, and the most invariant relative to different environment conditions and temperatures[14]. Several studies state than a detected temperature on the face of above 38°C is considered to be febrile[80]

4.4.1 Proposed Solution

Since we were estimating the temperature of a person through its eye corners temperature, the first obvious step was to localize the subject's face in the input image. As explained in section 4.6 , this step was solved by using our YOLO-based face detector, which returned a bounding box containing the face.

Our first approach was to simply get the 10 hottest pixels in the bounding box, after removing all temperature values outside of the [34-40]°C range (temperature equalization). These values were picked based on the average body temperature of a person, which is usually around 36.5°C, and also based on the body temperature in a febrile subject, which can go up to around 40°C. With this in mind, the selected range seemed like a good choice for our application. However, while testing this method on our recordings, it was discovered that in very hot days and when the car is out in the sun, temperatures inside the car could go up to 40°C and even more, so using the approach described above, the hottest points picked could be inside the facial bounding-box but outside of the face region. An example of this situation is shown in Figure 4.6.

To account for this, and because a thermal facial landmark detector that localized several facial points was already developed, including inner eye corner points, it was decided to use it to pinpoint the mentioned landmarks and then get the real temperature values in those locations. Since it was known that the landmark detector was not perfect, and sometimes the returned points did not correspond to the exact location of the eye inner corners (even though they would be close), it was decided to create a bounding box, using the lower bound of the eye corners y coordinate for the bottom coordinate, using both eyes x coordinates for left and right boundaries, and using the lower bound of the eyebrows corner y coordinate as the upper bound. To make sure that the bounding box included the eye corners, its limits were expanded by a pre-determined value. A block diagram representing the algorithm pipeline is presented in Figure 4.7

The idea of the bounding box was to make the system as robust as possible: this way, it could be always guaranteed that the eye inner corners would be present in the bounding box. After creating it, the 10 hottest pixels inside the described bounding box would be localized, and their temperature would be averaged to give the final result. In pretty much all the test situations, these pixels would be in the eye corners, as expected.

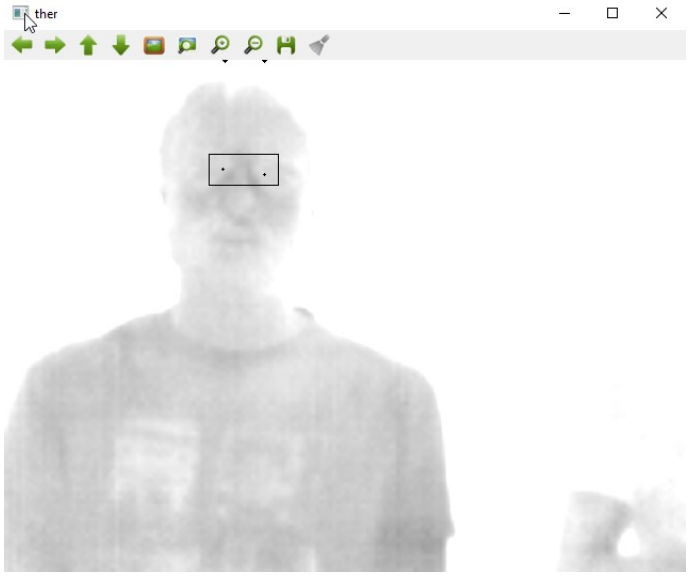

Figure 4.6: Thermal frame taken inside the car. The surroundings are hotter than the subject's body

Since body temperature is a variable that is not supposed to vary unpredictably, and to make the temperature measurement more robust, it was determined that the system would output a temperature measurement once every 10 frames: the estimation would take place in every frame, introduced in a 1x10 array, and when the 10th frame arrived, the average of the values of the mentioned array was calculated and given as final output, and the array would be emptied out to give place to another sequence. An image showing the output of the system (in console) is presented in Figure 4.8

4.4.2 Results

Due to lack of time, the accuracy of the thermal sensor could not be tested. Because of this, only the precision of the landmarks predicted and corresponding bounding box could be tested, and so this system depends exclusively on the accuracy of the face detector and the facial keypoint detector.

4.4.3 Conclusions

As said above, it became clear that body temperature estimation could not be carried out when the surrounding temperature was too high (in sunny days without air conditioner inside the car, for example). This is also stated in the "Deployment, implementation and operational guidelines for identifying febrile humans using a screening thermograph" standard[10], where it's said that body temperature estimation should only take place in a controlled environment, at about 20^{o}C.

Another issue occurs when the subject is wearing glasses: since most glasses cannot transmit wavelengths above 3 µm, and thermal cameras operate between the 8-14 µm wavelengths, they

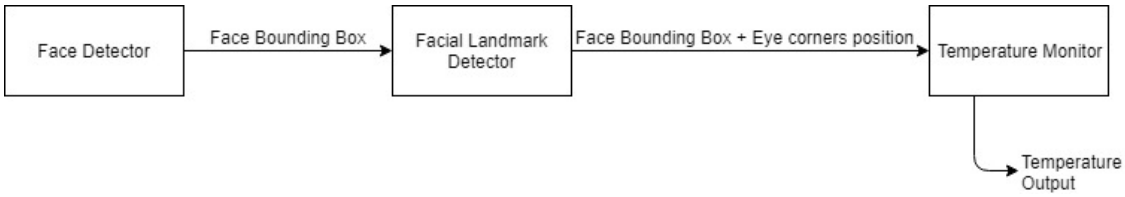

Figure 4.7: Block diagram describing the system

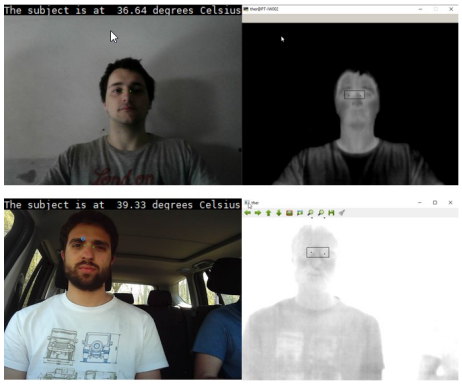

Figure 4.8: Images showing the output of the system

appear opaque in a thermal frame. This means that the eye corners do not appear in images where the person is wearing glasses and, therefore, we can't measure its temperature.

Another factor that should be investigated is regarding the used camera's accuracy: according to the specifications provided by the manufacturer in their website, the camera has an uncertainty of +-2^{o}C, which for an application such as temperature monitoring of a person, is a significant value. It is possible that this is a "worst-case" value, meaning that a temperature measurement right after the camera's thermal calibration can still be a valid approximation. However, the camera should be tested under different situations and environmental conditions, using for example a black-body at a fixed temperature to verify the camera's thermal precision. Also regarding thermal accuracy, we are relying on the camera's automatic calibration using its internal shutter, and we are not sure if it's worthy to use a more traditional approach to thermal calibration, even though we discovered a paper that clearly states that there is no advantage in using black bodies for thermal calibration of our camera, the FLIR ONE Pro[81]

4.5 Face Recognition

In this chapter, we'll detail the approach taken for the task of face recognition, specifically the tested approaches in the beginning(replicating the state of the art) followed by the presentation of our final solution. A technical review will be provided when we see fit. It should be noted that some advantages of using thermal images for face recognition is its robustness to different

light conditions (it works the same during the day as during the night) and also to prevent face recognition bypassing.

4.5.1 Approaches

Our first approaches primarily followed the state of the art, namely the usage of hand-crafted features proven[42] to be good for thermal images, including HOG, LBP and Gabor feature extraction (singularly or in conjunction). Other techniques tried included the popular and traditional eigenFaces[82] and FisherFaces[83] (using an openCV framework), and also using a pre-trained RGB FaceNet[39] feature extractor applied to equalized thermal images. However, we were never able to achieve the results described in some of the papers.

4.5.2 Proposed Solution

After trying out several approaches popular in the thermal community, we started trying out new methods that were state of the art in visible images, like Triplet Networks (FaceNet) and Siamese Networks. To train and test the network, we used our dataset together with the RGB-D-T dataset labelled with our method. Adding this dataset allowed us to have a total of 89 people, keeping 5 outside of training for other tests (test 2 and 3), instead of the 38 people that we currently have. We settled for the Siamese Network.

4.5.2.1 Technical Review

A Siamese neural network is a type of neural network architecture described by having two inputs following the same subnetwork (same layers and same weights). They are mainly utilized in tasks that involve finding similarity between objects that are similar or of the same class. Basically, each subnetwork works as a mapping between input space (images, in our case) to an embedding space where the euclidean distance between each feature representation directly corresponds to class similarity. As such, at the end of the feature extraction phase, we obtain two embeddings of two images, and the output of the network is their euclidean distance. Assuming our inputs are of the same object, it makes sense to use a similar model to process a similar input but trained to "discover" the features that best represent differences between classes.

To train the network, a contrastive loss is used, described by equation 4.1.

$$f(x) = (1-Y) * \frac{1}{2} * (Dw)^2 + (Y) * \frac{1}{2} * max(0, m - Dw)^2 \tag{4.1}$$

In the formula above, Dw refers to any distance function: in our case, we are using the euclidean distance, but it could be other, like for example cosine similarity. As we can see, the loss used will try to bring images of the same class together and place images from different classes far away. An image describing the network can be seen in 4.9

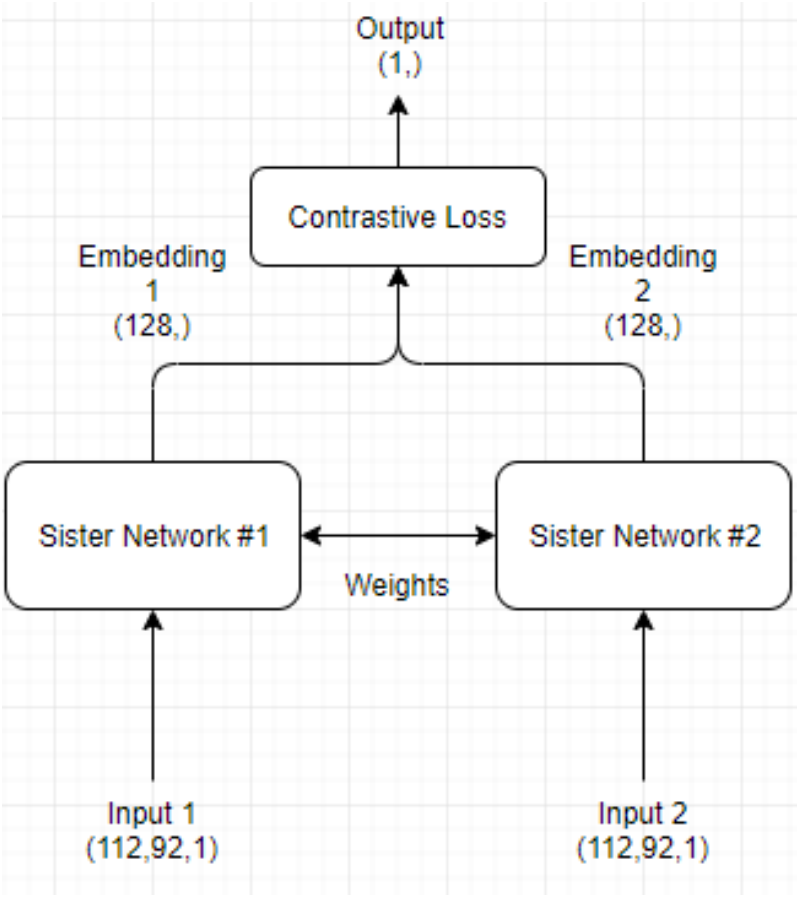

Figure 4.9: Siamese Network

4.5.2.2 Image pre-processing

As input to the Siamese network, grayscale images containing only the face bounding boxes were used, and resized to a common size (height = 112, width = 92): these values were picked keeping in mind the average height-to-width ratio of human faces so that we could minimize the distortion added with the resizing of pictures. To enhance their contrast, and as usually done along this thesis, histogram equalization was applied on the facial images, followed by normalizing the images (subtracting the image by its mean and further dividing it by its maximum). This normalization process is done to improve convergence speed and accuracy (by scaling the inputs) and to center the data so that vanishing and exploding gradients are attenuated.

An example of this pre-processing pipeline is showed in Figure 4.10

4.5.2.3 Feature extraction CNN

As it can be seen in Table 4.7, each input image goes through this feature extraction network, outputting its 128-feature embedding. After this process, the euclidean distance between both inputs is computed: in the training phase, this value is given as input to the loss function, along

Figure 4.10: Image pre-processing. Left image is the corresponding RGB image, middle image is the histogram-equalized thermal image and the right image is the input to the network, after normalization

with the corresponding label, and in the prediction phase a decision is made based on that output. To arrive at the presented final solution, the standard procedure employed along this thesis was followed: starting from a standard network, we aimed to minimize the validation loss. Regular convolutional blocks were used (Convolutional layer followed by Activation and Maxpooling), and measures to avoid overfitting were employed, like adding Batch Normalization and Dropout.

4.5.3 Results

Final training results are shown in Table 4.8. We can notice that the final training loss and final validation loss were similar, meaning that the network is not overfitting relative to the training images. We are pleased with the obtained results, since both losses reached a low value (random guess would correspond to a 0.5 loss). When we trained the model, and everytime we did it, we would split all of the available data in the beginning of the script, so that we would have 80% of all the images for training, 10% for the validation set and the remaining 10% for the test set.

4.5.3.1 Test 1: Who is this?

To perform this test, and to check if the model was working well and effectively approximated all image embeddings of the same person and add distance to image embeddings of different persons, we put aside 100 images for the test set. For each of the test images we selected one random image from each person from the training set (that should comprise our saved database). Then, for each test image, we ran the model for the input image with the random images selected from each person, and kept the lowest score.

As shown in Table 4.9, a top-1 accuracy metric of 95% was reached. Also, the average score of the right predictions was 0.054, the average score of the wrong predictions was 0.095 and the lowest wrong score was of 0.069. For more conclusive results, more test images shall be used. In theory, the perfect scenario would be to take into account all the images contained in the verification dataset.

4.5.3.2 Test 2: Is this person in the database?

An important feature to be included on the face recognition system was the ability to reject a non-authorized person that entered the car. Since the output of the system was a float value (the lower,

Type	***Filters***	***Size / Stride***	***Output***
Input Image	-	-	(112,92,1)
Convolutional	64	5x5 / 2	(54,44,64)
BatchNormalization	-	-	(54,44,64)
Activation (ReLU)	-	-	(54,44,64)
MaxPooling	-	3x3 / 2	(26,21,64)
Dropout (20%)	-	-	(26,21,64)
Convolutional	512	1x1	(26,21,512)
Activation (ReLU)	-	-	(26,21,512)
Fully-Connected	-	-	(512,)
Dropout (20%)	-		(512,)
Fully-Connected	-	-	(128,)
L2-Normalization	-	-	(128,)

Table 4.7: Feature extraction layers

	Training Loss	**Validation Loss**	***Batch Size***	**Loss Function**	**Optimizer and Learning Rate**	**Epochs**
Final model	0.0204	0.0196	8	Contrastive Loss	Adam(lr=0.01)	84

Table 4.8: Information about final model

k=1, N=86	#1
Top-1 accuracy	95%
Average Score	0.054
Wrong Score	0.095
Lowest Wrong Score	0.069

Table 4.9: Results for test 1

the more confident the system was of its guess), and in an effort to try and find a suitable threshold value for rejecting a person, we tested the system by trying to predict the three persons that were left out of the training dataset with the training dataset: this way, we would be checking how the system was reacting when that specific use case was happening, and we could analyze the returned score results. Naturally, there is no top-1 accuracy, since there is no way the network can guess who the person is without a saved face image of that same person.

As it can be seen in Table 4.10, the average score was 0.348, while the lowest score was 0.102. This suggests that a value of 0.1 can be a good threshold for person rejection: if any inference made is over 0.1, we reject the prediction, otherwise we can accept it. Still, more extensive tests should be performed to confirm this value.

4.5.3.3 Test 3: Can it differentiate between never seen people?

To see if this system was able to differentiate between persons that it not trained with, we used the same method as in test 1, but with a dataset of only 3 people that were left out of the training set.

k=3, N=86	Looking for not allowed people in a database
Average Score	0.348
Lowest Score	0.102

Table 4.10: Results for test 2

Results for this test are shown in Table 4.11. This is an expected result: analyzing these scores with the ones from test 1, the network is more confident in predicting over persons from the training set compared with persons from the test set. Still, it's a very interesting result that indeed the network learned how to distinguish faces in thermal images. Tests with more sujects should be performed in the future.

4.5.3.4 Test 4: Twins test

Fortunately, we had a set of twins in our dataset, and we thought it would be interesting to see if the model was able to correctly recognize them and distinguish between them. So, for this test, the procedure was similar to the first test, but we only used as input to the network images from them both and only compared with images from them both. 41 test images were used, and the best results are displayed in Table 4.12.

As shown in Table 4.12, the network was successful in distinguishing between twins, reaching an accuracy of 92.68% with a total of 41 images. This image should be redone with more test images for more proper conclusions, but it's still a valid result.

4.5.4 Conclusions

A very accurate face recognizer was developed during the course of this thesis, based exclusively on thermal images. Even though we are pleased with the obtained results, several improvements can still be made, regarding robustness to different persons and regarding speed.

The recognition procedure can be standardized, like for example the procedure adopted by Apple's faceID where the iPhone's owner can make its registration by recording a small video performing simple movements, like head rotations. Then, input images can be compared with the ones that were recorded for registration. Even though the test 1 provides good results and shows that indeed the network is learning to distinguish between all the used images of faces, it doesn't make sense to select only one (or k) random images to compare with the input image, since we are relying on randomness and we carry a risk of selecting an image where the person is completely sideways or looking down or another bad representation.

k=1, N=3	#1	#2	#3	#4	#5
Top-1 accuracy	96.5%	95.5%	95.0%	94.5%	94.0%
Average Score	0.274	0.244	0.252	0.254	0.257

Table 4.11: Results for test 3

Number of images per person	*Top-1 accuracy*
10	92.68%
5	85.36%

Table 4.12: Results for test 4

To make the system much faster, and because the decision process only involves calculating the euclidean distance between embeddings, it's possible to keep only the feature embeddings in the allowed people database, instead of the images. This would allow us to cut the network in half, keeping just one feature extraction network and to classify a person we could just calculate the euclidean distance between the network's output and the feature vectors saved in the database.

It should be noted that all the images used for testing were taken in the same day as the images from the training set, so we are not sure if the person's thermal face can appear different in a different day or in different temperature conditions. However, recording our dataset we try to account for this by raising/lowering the car's temperature while the person performs the script. Still, this situation could not be tested because that would imply recording the same person in different days and there was simply no possibilities to do that.

The ideal case would be that the network learned how to distinguish between faces that it has never seen. This situation was tested with only 3 persons, and the results were better than expected. Still, this should be tested with more subjects. We are pleased with our results, building a system capable of, in an example use case, determine who is the person inside the car from a given training database, and we believe that it is able to tell if a person is part of the database or not.

4.6 Fatigue and Drowsiness Monitoring

In this chapter, we'll detail the approach taken for the task of fatigue and drowsiness monitoring, specifically the attempted approaches in the beginning (in accordance with the state of the art) followed by the presentation of our final solution.

It's important to mention beforehand that traditional methods to analyze fatigue and drowsiness in a subject (using cameras) is to check for the state of the eye (if it's closed or opened), its blinking rate(measured in blinks per minute) and blinking total time(measured in seconds), if the person is yawning frequently and also checking for extreme head poses.

4.6.1 Approaches

As referred in the state-of-the-art section of this thesis, normal methods involve estimating the eye aspect ratio (EAR) (width to height aspect ratio) of the eye, to determine its state(open, closed and even semi-closed). To achieve this, one should be able to retrieve at least four points per eye(left and right corner, and top and bottom points), so that height and width of the eye could be

Figure 4.11: Extracted RGB eye images from our dataset

calculated. However, after starting this implementation, we soon realized that it could not be the best approach, due to the low resolution provided by our thermal camera and, consequently, not so precise facial landmarks extracted from our detector.

Other attempted methods included treating the problem as a binary classification task, for example by getting the image frame difference, like seen in related works: the purpose is that, when a blink occurs, the amount of black pixels in the evaluated region would go up, and by detecting this increase (using horizontal/vertical projection of the image, for example), a blink could be detected. However, none of these approaches were as good compared with our final proposed solution. With it, a system was developed capable of detecting if a person was sleeping and capable of detecting microsleeps, a fatigue criteria defined as brief, unintended episodes of prolonged eye closure.

4.6.2 Labelling the images

To be able to train/develop an algorithm capable of concluding if the eyes were opened or closed, labelled images of eye regions were needed. Since, as mentioned before, we had a camera capable of taking synchronized and aligned (by us) images on the RGB and T modalities, the RGB images were used to label the thermal ones, like for the facial regions and landmarks labelling, .

4.6.2.1 RGB labelling

For this to happen, a (very good) RGB eye state classificator was needed. To extract the eye patches, a script was developed that would iterate over all of our labelled images and would create and save an image containing only the eye region, using the left and right corners of the eyebrow and the left and right corners of the eye. An example image is presented in Figure 4.11.

After automatically generating these eye images, another script was developed, able to iterate over all the created images and with the press of a button, save them in a separate folder: one folder for the open eyes and another for the closed eyes. With these two folders, we now had labelled eye images to train and develop the algorithms that we wanted.

Type	***Filters***	***Size***	***Output***
Input Image	-	-	(50,50,3)
Convolutional	32	3x3	(48,48,32)
Activation (ReLU)	-	-	(48,48,32)
MaxPooling	-	2x2	(24,24,32)
Convolutional	32	3x3	(22,22,32)
Activation (ReLU)	-	-	(2,22,32)
MaxPooling	-	2x2	(11,11,32)
Convolutional	64	3x3	(9,9,64)
Activation (ReLU)	-	-	(9,9,64)
MaxPooling	-	2x2	(4,4,64)
Fully-Connected	-	-	(64,)
Activation (ReLU)	-	-	(64,)
Dropout (50%)	-	-	(64,)
Fully-Connected	-	-	(1,)

Table 4.13: Architecture of the RGB network

	Training Accuracy	**Validation Accuracy**	***Batch Size***	**Loss Function**	**Optimizer and Learning Rate**	**Epochs**
Final model	99.7%	99.4%	16	Binary Cross-Entropy	Adam(lr=0.001)	18

Table 4.14: Results for the RGB convolutional neural network

4.6.2.2 RGB CNN

For the RGB domain, a regular CNN was developed using Keras, containing three standard convolutional blocks (composed by convolutional layer->activation with ReLU->max pooling) for feature extraction, followed by a fully-connected layer. A 50% dropout layer was added after the fully-connected layer, with the purpose of reducing overfitting. The output of the network is a number between 0 and 1: 0 represents opened eye, and 1 represents closed eye. Since we were only using two labels, the choice for the loss function was binary cross-entropy[84], and the chosen optimizer was Adam[85]. With this architecture, and with around 1000 images for training, an accuracy of around 99% in the validation and test set was reached.

4.6.3 Proposed Solution

Developing this neural network on the RGB domain allowed us to automatically label images to train a CNN for the thermal modality. So, instead of hand labelling(classifying) like done before, we replaced the classificator (previously, the author) with the trained algorithm (CNN described above): this way, we could iterate over all the images from our dataset, extract the corresponding eye areas in both domains, perform classification with the trained RGB classificator, consequently labelling the matching thermal image and saving it to the corresponding folder (opened eye or closed eye). Because, in thermal frames, glasses (normal glasses and sunglasses) block the eye

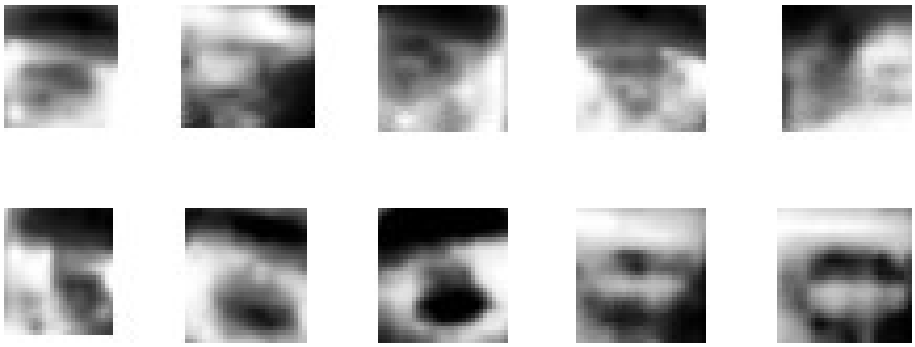

Figure 4.12: Equalized thermal images of opened eyes and closed eyes

regions, they were filtered out using the labels available in our info.csv file, which contained information about the existence of glasses in each frame.

With this procedure, we were able to extract exactly 6965 images of opened eyes and 1901 images of closed eyes. We now had all the necessary conditions to train our thermal CNN.

4.6.3.1 Image pre-processing and data augmentation

We decided to pre-process all images with histogram equalization to enhance their contrast and because we were getting better results, compared with using no equalization. As stated in section 4.2, glasses do not transmit infrared radiation, so if a person is wearing glasses we are not able to estimate its eye state. In those cases, the only solution would be to use the RGB eye classificator or simply not performing the classification. This situation was easily identifiable by running our glasses detector. A similar approach as in section 4.5 was adopted to normalize the input images, subtracting them by their mean and dividing them by their maximum pixel value. In each batch that was served as input to the network during training, 50% of those images were augmented with random rotations and translations, blurs, gaussian noise and also contrast and color modifications.

As expected, left eye images and right eye images appear to be very similar, but they are flipped around the vertical axis. As an attempt to "normalize" the data, and because we were applying the same predictor to both eyes, we decided to arbitrarily apply a vertical flip to the left eye images, so that it would look very much alike to the right eye images. Obviously, we had to always flip the left eye region before using the model to predict eye state.

4.6.3.2 Thermal CNN

A CNN was trained with the purpose of, given an histogram-equalized eye cropped image, classifying it as opened or closed. Starting with the previous RGB architecture, it was noticed that the network didn't work as well, which is natural given the increased difficulty of the task. In Figure 4.12 images of both eye states are shown, where it can be seen that, even for a human, the exercise is not easy at all.

Type	***Filters***	***Size***	***Output***
Input Image	-	-	(50,50,1)
Convolutional	512	5x5	(46,46,512)
BatchNormalization	-	-	(46,46,512)
Activation (ReLU)	-	-	(46,46,512)
MaxPooling	-	2x2	(23,23,512)
Convolutional	512	3x3	(21,21,512)
BatchNormalization	-	-	(21,21,512)
Activation (ReLU)	-	-	(21,21,512)
MaxPooling	-	2x2	(10,10,512)
Convolutional	1024	3x3	(8,8,1024)
Convolutional	2048	1x1	(8,8,2048)
BatchNormalization	-	-	(8,8,2048)
Activation (ReLU)	-	-	(8,8,2048)
MaxPooling	-	2x2	(4,4,2048)
Fully-Connected	-	-	(64,)
Activation (ReLU)	-	-	(64,)
Fully-Connected	-	-	(1,)

Table 4.15: Thermal architecture

To account for this, and since we now had a significant amount of training images, it was decided to make some changes in order to make the network deeper. Starting from the RGB architecture, we followed a trial-and-error approach, with the objective of maximizing the accuracy and minimizing the loss in the validation and test sets. After experimenting out with different architectures and layers, we arrived at our final solution: using the standard procedure used along this thesis to separate the data (randomly dividing all our images in 80% for training, 10% for validation and 10% for testing at the start of each training script), we kept the model that minimized our validation loss (same loss used as in the RGB case, binary cross-entropy) and that maximized our testing accuracy (in the end of the training step, the script would automatically use the trained model to test on the testing set, outputting the accuracy achieved). Relative to the RGB architecture, we replaced the dropout layer that was present after the 64 neurons fully-connected layer with batch normalization layers after each convolutional layer, the number of filters used in the convolutional layers was increased and L2-Normalization was added to the fully-connected layer.

Another adopted measure was to make sure that, for every batch that was fed to the network in each step, an equal amount of closed eye images and opened eye images was provided: because we had about 3.5 times more opened eye images than closed eye images (unbalanced classes), we were worried that the network could become much better at predicting one class relative to the other, so by doing this we guaranteed that in total the network iterated through the same number of closed and opened eye images.

	Training Accuracy	**Validation Accuracy**	***Batch Size***	**Loss Function**	**Optimizer and Learning Rate**	**Epochs**
Final model	95.06%	91.00%	32	Binary Cross-Entropy	SGD(lr=0.001)	80

Table 4.16: Training information

	Subject 1	Subject 2	Subject 3	Author	Machine
Accuracy	65%	60%	61%	58%	91%

Table 4.17: Test results

4.6.4 Results

Following the procedure described in the previous section, we retrieved the best model, achieving a validation accuracy and a test accuracy of 91%.

4.6.4.1 Human vs Machine test

To assess our algorithm, we decided to perform a test where we compare the predictions of humans with the predictions from our model. Four persons, including two subjects that are currently writing a thesis on thermal images, were shown 100 thermal images containing closed or opened eyes and tried to guess the displayed eye state. The results are shown on Table 4.17

As we can verify on Table 4.17, humans achieved an average accuracy of 61%. This corresponds to our expectations before the test: in some images, it's possible to notice the eye contours and generally conclude if the eye is opened or not, whereas in other images it's basically random. We were pleased with the obtained results, showing that it's possible to develop a neural network model capable of learning a task that is difficult for humans with satisfactory results. However, we should keep in mind that, unlike the tested persons, the model has seen several images of that type.

4.6.4.2 Microsleep Detection

As described in the dataset creation section 3.5, there is an activity in our dataset where the subjects are asked to close their eyes for five seconds, effectively counting as a microsleep (having eyes closed for more than one second). As such, a meaningful test to perform was to check if our system was able to detect that during that time period, the person closed his/her eyes. Six subjects were then labelled as having their eyes closed or opened during that period, and the test was performed.

On Table 4.17, the results for this experience can be seen. The ground truth labels are for both eyes, and on those frames we made sure that both eyes had the same state. Since we were making predictions for one eye only, the results are presented with the prediction for the left eye and also for the right eye. In some of the annotated images, the eye state is a bit ambiguous in the sense that, when the eye is almost closed it can have an "opened" label or a "closed" label. In those cases, if the algorithm got the answer wrong, we would mark it with a checkmark with a cross.

	#1	#2	#3	#4	#5	#6	#7	#8
Subject 26	open	closed	closed	closed	closed	closed	opened	opened
Left Eye Prediction	✓	✗	✓	✓	✓	✓	✓	✓
Right Eye Prediction	✓	✓	✓	✓	✓	✓	✓	✓
Subject 27	open	closed	closed	closed	closed	closed	opened	-
Left Eye Prediction	✓	✓✗	✓✗	✓	✓	✓	✓	
Right Eye Prediction	✓	✓✗	✓✗	✓	✓	✓	✓	
Subject 28	open	closed	closed	opened	-	-	-	-
Left Eye Prediction	✓	✓	✓	✓				
Right Eye Prediction	✓	✓	✓	✓				
Subject 29	open	closed	closed	closed	closed	opened	-	-
Left Eye Prediction	✓	✗	✗	✗	✗	✓		
Right Eye Prediction	✓	✗	✗	✗	✗	✓		
Subject 30	opened	closed	closed	closed	closed	opened	-	-
Left Eye Prediction	✓	✓	✗	✗	✓	✓		
Right Eye Prediction	✗	✗	✗	✗	✓	✓		
Subject 31	opened	closed	closed	closed	closed	closed	opened	-
Left Eye Prediction	✓	✓	✓	✓	✓	✓	✓	
Right Eye Prediction	✓	✓	✓	✓	✓	✓	✓	

Table 4.18: Microsleep detection. In some situations a checkmark with a cross was used, because the image was ambiguous

We can see that, for some subjects the predictions are 100% right, and for others the results are not very satisfactory. The overall accuracy, removing predictions with checkmarks and crosses, was 77%.

4.6.5 Conclusions

In this section, the eye state classifier developed during the course of this thesis was described. Overall, we are happy with the results, building a model able to predict above human level and robust to all light conditions. As future work, the network architecture can be changed and perfected, so that higher accuracies are reached. Also, the dataset can be enlarged and cleaned, since for now we are automatically labelling it with an RGB classifier trained by us, so we are sure there are labels that are not correctly assigned. Another measure could be to label the data not as to perform a binary classification, but with more descriptive labels, for example a number between 0 and 1 directly related to the percentage of eye openness, because with our method there are a lot of ambiguous cases where the eye is almost closed but not entirely.

Regarding the fatigue and drowsiness detection case, a method to detect yawns and to detect extreme head poses can be developed, since these two situations are also criteria for this use case that weren't explored.

	Test 1	Test 2	Test 3	Test 4	Test 5
Top-1 Accuracy	96.5%	95.5%	95.0%	94.5%	94.0%
Confidence	0.274	0.244	0.252	0.254	0.257

Table 4.19: Test 2: Unseen dataset results

	Test 3 - Looking for not allowed people in a database
Average Score	0.348
Lowest Score	0.102

Table 4.20: Test 3

	Test 1	Test 2	...
Top-1 accuracy	95%	-	-
Average Confidence	0.054	-	-
Wrong Confidence	0.095	-	-
Lowest Wrong Confidence	0.069	-	-

Table 4.21: Test 1: All dataset

Chapter 5

Conclusions and Future Work

The main objectives proposed for this dissertation were accomplished. A system capable of monitoring people inside the vehicle using exclusively thermal images was developed. However, several improvements can still be done.

5.1 Face Detection

The face detector was successfully implemented, being capable of running in real time and with very high precision. As future work, more tests should be done with different people and with different conditions. Also, even though we believe that the occupancy use case (being able to detect how many faces are inside the vehicle) will work, this situation was not tested because we do not have images to test it.

5.2 Glasses Detector

The glasses detector was not an initial objective of this thesis, but the need came up to determine if a person was wearing glasses or not. This subsystem was successfully implemented, capable of running only with a facial bounding box input as we wanted, in real time and with high accuracies. As future work for this subsystem, higher accuracies can be achieved by training the network with more images and with different persons and glasses.

5.3 Facial Landmark Detector

A facial landmark predictor was created, using an ensemble of regression trees. This subsystem achieved good results, although they could be better: for frontal faces it works very well, but when the faces are rotated or looking sideways the system drops its accuracy. Through the course of this thesis, other implementations were tried based on convolutional neural networks, but due to lack of time we were not able to correctly optimize them. As future work, we would like to check if these implementations could achieve a lower error than our current facial keypoint detector.

5.4 Face Recognizer

A siamese neural network was developed to perform the task of face verification. The results were very good, demonstrating that these kinds of networks are a good solution for this exercise. As future work, we would like to develop a network similar to FaceNet, capable of determining if the input images correspond to the same person or not but without training the network with the tested persons. To achieve this, we believe that a much higher number of persons was needed to train the network: even though we used 89 people to train it (using the rgb-d-t dataset and ours), FaceNet was trained with datasets containing a total of 1680 (Labeled Faces in the Wild) and 1595 (Youtube Faces Database).

5.5 Temperature Monitor

A system able to estimate human body temperature by measuring the temperature of the inner corners of the eyes was created. Algorithm-wise, this system depends exclusively on the facial keypoint detector, so improving the keypoint detector will result in an improvement on the temperature monitor. In terms of thermal accuracy, the system was not tested properly so we're not sure that this application is viable as it is. As future work, tests should be performed to determine the camera's thermal accuracy and if anything can be done in order to improve it, such as using a in-scene blackbody for the thermal calibration.

5.6 Fatigue and Drowsiness Monitoring

A classifier able to tell if a person's eyes are opened or closed was developed, achieving accuracies that are above human level. Several other criteria for fatigue and drowsiness detection were not explored, such as yawn detection, blinking frequency calculation (even though this could be achieved with our current eye state classifier) and extreme head poses estimation. As future work, these criteria could be explored. Also, the accuracy of our current eye state classifier can still be improved, by training it with more images and maybe changing the current architecture.

Appendix A

Capture Guide

From
BrgP/ENG-P
BrgP/ENG-P

Our Reference
Gustavo Silva
Rui Monteiro

Tel

Braga
19 March 2018

Report
Issue Capture guide
Topic

1 Equipment

Equipment	Quantity	Model	Purpose
Respiratory measurement system	1	?	Ground truth data for evaluation.
Heart rate measurement system	1	Quirumed portable pulse oximeter (https://www.quirumed.com/en/portable-pulse-oximeter-pulse-and-oxygen-saturation.html?sid=46314)	
RGB + Thermal camera	1	FLIR ONE Pro	Record synchronized thermal and visible video
Cigarettes	40+		Simulating smoking activities
Lighter	1		Light cigarettes
Air temperature sensor	1		Register the ambient temperature (not mandatory)
Computer with Linux and available USB port	1		Obtain frames from camera

USB-C female to USB-A male cable	1		Connect the camera to the computer
Glasses/sunglasses	3+	Any (preferably different in shape)	Simulate situations where the occupants wear glasses
Camera mount	1		Hold the camera firmly in place
Timer	1	e. g. smartphone	Ensure correct activity times during recording

2 Setup

- Obtain written consent from the subjects.
- Explain the process to the subjects.
- Choose a route or multiple routes that will be taken. The route should take slightly more than 7min and 10s to traverse, though this is not a requirement. We don't need every subject to perform the route, some subjects can be recorded when the car is not moving. We just need to check for unexpected issues that may occur while the car is moving on the road.
- Connect the camera to the Jetson TX2.
- Secure the camera in front of the right front seat.

3 Annotation

The recommended software to be used to annotate bounding boxes and points in extracted frames is available at https://github.com/NaturalIntelligence/imglab under the MIT license.
A facial bounding box should be annotated for all visible faces. Its top should be delimited by the boundary between the forehead and the hair. The bottom corresponds to the limit of the chin. Left and right boundaries are defined to end exactly in front of the ears, therefore excluding them.
For each face, some important fiducial key points should also be annotated, as shown in the image.

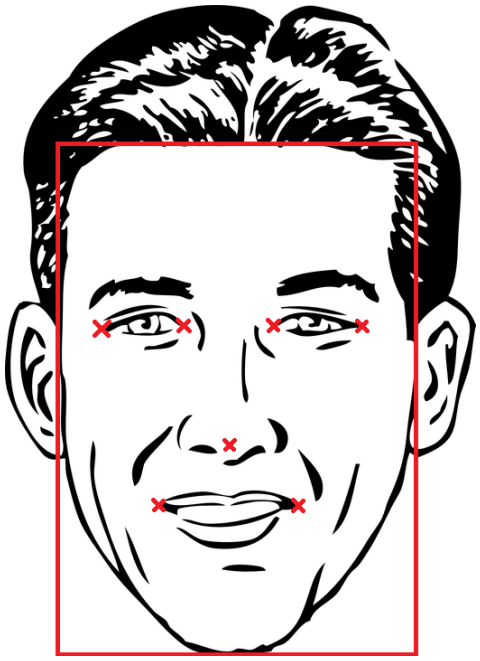

Figure 1 - Example of face annotation.

4 Data storage format

4.1 *Folder structure*

The root folder of the dataset should contain a file "demographics.csv" with information on each subject and a folder for each of them, named by their ID.

These folders contain multiple information:

- The visible component of the captured videos as a sequence of JPEG images, named by the ID of the frame.
- The thermal component of the captured videos as a sequence of 16bpp BMP images, named by the ID of the frame.
- A file "info.csv" with information on some of the extracted frames.
- A file "annotations.xml" with the annotations of the same extracted frames in Dlib's format.

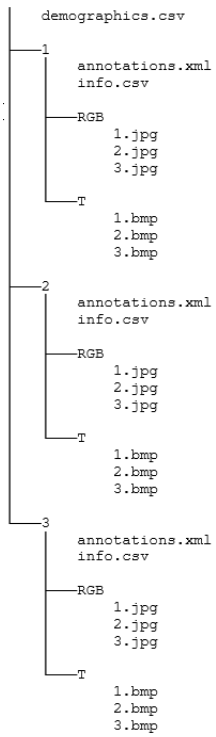

Figure 2 - Example folder organization

4.2 *Info.csv*

This file stores information on some (not all) of the recorded frames. The frame can be specified either by the corresponding time of by the frame ID.

time	frame	glasses	smoking	hot	expression	yawning	eye status	heart rate
520		yes	no	no	neutral	no	open	
	50	yes	no	no	neutral		closed	
	100	no	yes	no	happy	no		60
	115	yes	no	no	neutral			59
	120	yes	no	no	neutral			
	150	yes	no	no	neutral			
	180	yes	no	no	neutral		open	
	200	yes	no	no	neutral	yes	squinted	

Figure 3 - Example info file

4.3 *demographics.csv*

The demographics file stores information on some characteristics of each subject. If, in any field, the information is unknown, it should be left empty. The fields are:

- Person – numerical identification of the person
- Age
- Gender - Male, Female or Other
- Height and waist-to-head Height - in centimetres.
- Beard - yes/no.
- Moustache - yes/no.
- Race - one of:
    - white
    - asian
    - black
    - hispanic

    - other
- Hair size - one of:
    - short
    - medium
    - long

Person	Age	Gender	Height	Beard	Moustache	Race	Hair Size
1	18	Male	180	yes	yes	White	short
2	33	Female	163	no	no	Asian	medium
3	24	Male	176	no	yes	Black	short
4	68	Female	168	yes	no	Hispanic	long

Figure 4 - Example demographics file

5 Collection process

During recording, the temperature shall be changed once for each subject, in order to capture images in hot and cold environments for everyone. To minimize waiting times, the temperature should only be changed once per subject.

If at any "wait" instruction the required time has already passed, the process should be continued anyway without waiting.

When a step requires to "change vehicle temperature" the temperature should be set to the opposite of the current status. Therefore, if the vehicle is cold, the windows should be closed and the A/C set to maximum. Otherwise, the windows should be opened and the A/C set to minimum.

During the recording, the person responsible for the capture process should be in the backseat of the car, appearing in the footage and taking note of information regarding some of the frames.

Assuming a route is already defined, the steps for recording should be as follows:

1. Find a driver and a passenger subject (front seat) and collect information, to be stored in file "demographics.csv":
    a. ID
    b. Age
    c. Gender
    d. Height
    e. Waist-to-Head Height
    f. Beard
    g. Moustache
    h. Race
    i. Hair size
2. Remove glasses from subject.
3. Start capturing and start the clock, ensuring all devices are synchronized (start at the same time).
4. Subject enters the vehicle.
5. Start driving.
6. If windows are open, willing subjects light up a cigarette and start smoking. The cigarette should be lighted directly from the mouth using a lighter. From time to time place the

cigarette outside the vehicle (arm outside the window). Subjects should keep smoking during the following steps.

7. Wait until 00:30.
8. Perform head movements, with the following timings:
    a. ~1 second to adopt the final position;
    b. ~1 second holding in final position;
    c. ~1 second going back to initial position;
    d. ~1 second waiting time before going for next position.

    The movements should be done in the following order:
    1. Roll to the left (roll)
    2. Roll to the right (roll)
    3. Rotate to the left (yaw)
    4. Rotate to the right (yaw)
    5. Rotate down (pitch)
    6. Rotate up (pitch)
9. Wait until 01:10.
10. Perform the following facial expressions, ensuring at least 2 seconds per expression and a minimum waiting time between expressions of at least 5 seconds:
    a. Happy
    b. Sad
    c. Surprise
    d. Fear
    e. Anger
    f. Disgust
11. Wait until 02:10.
12. Perform fatigue movements, with a minimum waiting time between movements of at least 2 seconds.
    a. Blink in each second for ~5 seconds
    b. Yawn for ~3 seconds
    c. Close eyes for ~5 seconds
    d. Squint for ~5 seconds
13. Wait until 2:50.
14. Wear glasses.
15. Change vehicle temperature.
16. If windows are open, willing subjects light up a cigarette and start smoking. The cigarette should be lighted directly from the mouth using a lighter. From time to time place the cigarette outside the vehicle (arm outside the window).

17. Wait until 4:50.
18. Stop smoking.
19. Perform head movements, with the following timings:
    a. ~1 second to adopt the final position;
    b. ~1 second holding in final position;
    c. ~1 second going back to initial position;
    d. ~1 second waiting time before going for next position.

    The movements should be done in the following order:
    7. Roll to the left (roll)
    8. Roll to the right (roll)
    9. Rotate to the left (yaw)
    10. Rotate to the right (yaw)
    11. Rotate down (pitch)
    12. Rotate up (pitch)
20. Wait until 05:30.
21. Perform the following facial expressions, ensuring at least 2 seconds per expression and a minimum waiting time between expressions of at least 5 seconds:
    a. Happy
    b. Sad
    c. Surprise
    d. Fear
    e. Anger
    f. Disgust
22. Wait until 06:30.
23. Perform fatigue movements, with a minimum waiting time between movements of at least 2 seconds.
    a. Blink in each second for ~5 seconds
    b. Yawn for ~3 seconds
    c. Close eyes for ~5 seconds
    d. Squint for ~5 seconds
24. Wait until 07:10.
25. Continue driving until destination (if not already reached).
26. Passengers leave the vehicle.
27. Stop capturing and reset clock.
28. Stop capturing or repeat from step 1 with new subjects.

References

[1] Thermal images. URL: https://www.eeginfo-europe.com/neurofeedback/support/technical-infos-faq/thermal-images.html.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[3] Euclides Arcoverde, Rafael M. Duarte, Rafael M. Barreto, Joao Paulo Magalhaes, Carlos C. M. Bastos, Tsang Ing Ren, and George Cavalcanti. Enhanced real-time head pose estimation system for mobile device. 21:281–293, 01 2014.

[4] Marcin Kopaczka, Nikolai Blanik, Michael Czaplik, Nadine Carmen Hochhausen, Michael Paul, Carina Barbosa Pereira, Steffen Leonhardt, Vladimír Blazek, and Dorit Merhof. A thermal Infrared Face Database and Active Appearance Model Based Face Detection in a System for Pain Assessment in Sedated Patients. In *[Proceedings of the 11th German-Russian Conference on Biomedical Enginerering]*, pages 33–36, Aachen, Jun 2015. 11th German-Russian Conference on Biomedical Engineering, Aachen (Germany), 17 Jun 2015 - 19 Jun 2015. URL: https://publications.rwth-aachen.de/record/564363.

[5] What's new in yolo v3? URL: "https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b".

[6] Human error causes 94 percent of car accidents. URL: "https://blog.lawinfo.com/2017/09/06/human-error-causes-94-percent-of-car-accidents/".

[7] Road crash statistics. URL: "http://asirt.org/Initiatives/Informing-Road-Users/Road-Safety-Facts/Road-Crash-Statistics".

[8] Paul Michael Jerem. *Body surface temperature as an indicator of physiological state in wild birds*. PhD thesis, University of Glasgow, 2017.

[9] Glenn J. Tattersall. Infrared thermography: A non-invasive window into thermal physiology. *Comparative Biochemistry and Physiology Part A: Molecular Integrative Physiology*, 202:78 – 98, 2016. Special Issue: Ecophysiology methods: refining the old, validating the new and developing for the future. URL: http://www.sciencedirect.com/science/article/pii/S1095643316300435, doi:https://doi.org/10.1016/j.cbpa.2016.02.022.

[10] Medical electrical equipment – deployment, implementation and operational guidelines for identifying febrile humans using a screening thermograph. URL: https://www.iso.org/standard/69347.html.

[11] H Budzier and G Gerlach. Calibration of uncooled thermal infrared cameras. *Journal of Sensors and Sensor Systems*, 4(1):187–197, 2015.

[12] B.B. Lahiri, S. Bagavathiappan, T. Jayakumar, and John Philip. Medical applications of infrared thermography: A review. *Infrared Physics Technology*, 55(4):221 – 235, 2012. URL: http://www.sciencedirect.com/science/article/pii/S1350449512000308, doi:https://doi.org/10.1016/j.infrared.2012.03.007.

[13] Sebastian Budzan and Roman Wyżgolik. Face and eyes localization algorithm in thermal images for temperature measurement of the inner canthus of the eyes. *Infrared Physics & Technology*, 60:225–234, 2013.

[14] EFJ Ring, A Jung, J Zuber, P Rutowski, B Kalicki, and U Bajwa. Detecting fever in polish children by infrared thermography. In *Proceedings of the 9th International Conference on Quantitative Infrared Thermography*, volume 2, 2008.

[15] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[16] Arwa M Basbrain, John Q Gan, and Adrian Clark. Accuracy enhancement of the viola-jones algorithm for thermal face detection. In *International Conference on Intelligent Computing*, pages 71–82. Springer, 2017.

[17] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[18] Krishna Rao Kakkirala, Srinivasa Rao Chalamala, and SantoshKumar Jami. Thermal infrared face recognition: A review.

[19] HI Ashiba, HM Mansour, MF El-Kordy, and HM Ahmed. A new approach for contrast enhancement of infrared images based on contrast limited adaptive histogram equalization. *Applied Mathematics & Information Sciences Letters*, 3(3):123–125, 2015.

[20] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[21] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[22] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[23] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006.

[24] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.

[25] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[26] Xudong Sun, Pengcheng Wu, and Steven CH Hoi. Face detection using deep learning: An improved faster rcnn approach. *arXiv preprint arXiv:1701.08289*, 2017.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[29] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[30] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. 04 2018.

[33] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. URL: http://arxiv.org/abs/1708.02002, arXiv:1708.02002.

[34] Yuen Kiat Cheong, Vooi Voon Yap, and Humaira Nisar. A novel face detection algorithm using thermal imaging. In *Computer Applications and Industrial Electronics (ISCAIE), 2014 IEEE Symposium on*, pages 208–213. IEEE, 2014.

[35] Yufeng Zheng. Face detection and eyeglasses detection for thermal face recognition. In *Image Processing: Machine Vision Applications V*, volume 8300, page 83000C. International Society for Optics and Photonics, 2012.

[36] Arwa M. Basbrain, John Q. Gan, and Adrian Clark. Accuracy enhancement of the viola-jones algorithm for thermal face detection. In De-Shuang Huang, Abir Hussain, Kyungsook Han, and M. Michael Gromiha, editors, *Intelligent Computing Methodologies*, pages 71–82, Cham, 2017. Springer International Publishing.

[37] Chengjun Liu and Harry Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476, 2002.

[38] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[39] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[40] Goutam Majumder and Mrinal Kanti Bhowmik. Gabor-fast ica feature extraction for thermal face recognition using linear kernel support vector machine. In *Computational Intelligence and Networks (CINE), 2015 International Conference on*, pages 21–25. IEEE, 2015.

[41] Naser Zaeri, Faris Baker, and Rabie Dib. Thermal face recognition using moments invariants. In *SCIEI International Conference on Digital Signal Processing (ICDSP 2014)*, 2014.

[42] Gabriel Hermosilla, Javier Ruiz-del Solar, Rodrigo Verschae, and Mauricio Correa. A comparative study of thermal face recognition methods in unconstrained environments. *Pattern Recognition*, 45(7):2445–2459, 2012.

[43] Shutao Li, Dayi Gong, and Yuan Yuan. Face recognition using weber local descriptors. *Neurocomputing*, 122:272–283, 2013.

[44] Zhan Wu, Min Peng, and Tong Chen. Thermal face recognition using convolutional neural network. In *2016 International Conference on Optoelectronics and Image Processing (ICOIP)*, pages 6–9, June 2016. doi:10.1109/OPTIP.2016.7528489.

[45] Javier Ruiz-del Solar, Rodrigo Verschae, Gabriel Hermosilla, and Mauricio Correa. Thermal face recognition in unconstrained environments using histograms of lbp features. In *Local Binary Patterns: New Variants and Applications*, pages 219–243. Springer, 2014.

[46] Gabriel Hermosilla, Gonzalo Farias, Cesar San Martin, and Francisco Gallardo. Study of local matching-based facial recognition methods using thermal infrared imagery. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(08):1556012, 2015.

[47] Ayan Seal, Debotosh Bhattacharjee, Mita Nasipuri, Consuelo Gonzalo-Martin, and Ernestina Menasalvas. Fusion of visible and thermal images using a directed search method for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(04):1756005, 2017.

[48] Mohammad Hanif and Usman Ali. Optimized visual and thermal image fusion for efficient face recognition. In *Information Fusion, 2006 9th International Conference on*, pages 1–6. IEEE, 2006.

[49] Ayan Seal, Debotosh Bhattacharjee, and Mita Nasipuri. Human face recognition using random forest based fusion of à-trous wavelet transform coefficients from thermal and visible images. *AEU - International Journal of Electronics and Communications*, 70(8):1041 – 1049, 2016. URL: http://www.sciencedirect.com/science/article/pii/S1434841116301418, doi:https://doi.org/10.1016/j.aeue.2016.04.016.

[50] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[51] Gabriel Hermosilla, Francisco Gallardo, Gonzalo Farias, and Cesar San Martin. Fusion of visible and thermal descriptors using genetic algorithms for face recognition systems. *Sensors*, 15(8):17944–17962, 2015.

[52] Pierre Buyssens and Marinette Revenu. Fusion levels of visible and infrared modalities for face recognition. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–6. IEEE, 2010.

[53] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):607–626, 2009.

[54] CV Hari and Praveen Sankaran. Embedding vehicle driver face poses on manifolds. In *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2017 IEEE International Conference on*, pages 1–5. IEEE, 2017.

[55] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[56] K. Fornalczyk and A. Wojciechowski. Robust face model based approach to head pose estimation. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1291–1295, Sept 2017. doi:10.15439/2017F425.

[57] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. *CoRR*, abs/1710.00925, 2017. URL: http://arxiv.org/abs/1710.00925, arXiv:1710.00925.

[58] Gareth J Edwards, Timothy F Cootes, and Christopher J Taylor. Face recognition using active appearance models. In *European conference on computer vision*, pages 581–595. Springer, 1998.

[59] Marcin Kopaczka, Kemal Acar, and Dorit Merhof. Robust facial landmark detection and face tracking in thermal infrared images using active appearance models, 01 2016.

[60] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Deepgaze ii: Reading fixations from deep features trained on object recognition. *arXiv preprint arXiv:1610.01563*, 2016.

[61] R. Oyini Mbouna, S. G. Kong, and M. G. Chun. Visual analysis of eye state and head pose for driver alertness monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1462–1469, Sept 2013. doi:10.1109/TITS.2013.2262098.

[62] S. Darshana, D. Fernando, S. Jayawardena, S. Wickramanayake, and C. DeSilva. Efficient perclos and gaze measurement methodologies to estimate driver attention in real time. In *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, pages 289–294, Jan 2014. doi:10.1109/ISMS.2014.56.

[63] B. Mandal, L. Li, G. S. Wang, and J. Lin. Towards detection of bus driver fatigue based on robust visual analysis of eye state. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):545–557, March 2017. doi:10.1109/TITS.2016.2582900.

[64] Wenhui Dong and Xiaojuan Wu. Fatigue detection based on the distance of eyelid. In *Proceedings of 2005 IEEE International Workshop on VLSI Design and Video Technology, 2005.*, pages 365–368, May 2005. doi:10.1109/IWVDVT.2005.1504626.

[65] Y. Kurylyak, F. Lamonaca, and G. Mirabelli. Detection of the eye blinks for human's fatigue monitoring. In *2012 IEEE International Symposium on Medical Measurements and Applications Proceedings*, pages 1–4, May 2012. doi:10.1109/MeMeA.2012.6226666.

[66] J. Batista. A drowsiness and point of attention monitoring system for driver vigilance. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 702–708, Sept 2007. doi:10.1109/ITSC.2007.4357702.

[67] A. Simić, O. Kocić, M. Z. Bjelica, and M. Milošević. Driver monitoring algorithm for advanced driver assistance systems. In *2016 24th Telecommunications Forum (TELFOR)*, pages 1–4, Nov 2016. doi:10.1109/TELFOR.2016.7818908.

[68] Shangfei Wang, Zhilei Liu, Siliang Lv, Yanpeng Lv, Guobing Wu, Peng Peng, Fei Chen, and Xufa Wang. A natural visible and infrared facial expression database for expression recognition and emotion inference. 12:682 – 691, 12 2010.

[69] O. Nikisins, K. Nasrollahi, M. Greitans, and T. B. Moeslund. Rgb-d-t based face recognition. In *2014 22nd International Conference on Pattern Recognition*, pages 1716–1721, Aug 2014. doi:10.1109/ICPR.2014.302.

[70] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88, 5 2016. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC.

[71] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016. doi:10.1109/LSP.2016.2603342.

[72] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[73] C. Shavers, R. Li, and G. Lebby. An svm-based approach to face detection. In *2006 Proceeding of the Thirty-Eighth Southeastern Symposium on System Theory*, pages 362–366, March 2006. doi:10.1109/SSST.2006.1619082.

[74] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[75] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[76] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1685–1692, 2014.

[77] Benjamin S. Riggan, Nathaniel J. Short, and Shuowen Hu. Thermal to visible synthesis of face images using multiple regions. *CoRR*, abs/1803.07599, 2018. URL: http://arxiv.org/abs/1803.07599, arXiv:1803.07599.

[78] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009.

[79] Wei-Yin Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329–348, 2014.

[80] Eddie Y.K Ng, G.J.L Kawb, and W.M Chang. Analysis of ir thermal imager for mass blind fever screening. *Microvascular Research*, 68(2):104 – 109, 2004. URL: http://www.sciencedirect.com/science/article/pii/S0026286204000548, doi:https://doi.org/10.1016/j.mvr.2004.05.003.

[81] Allen R. Curran, Mark Klein, Mark L Hepokoski, and Corey Packard. Improving the accuracy of infrared measurements of skin temperature. In *Extreme Physiology Medicine*, 2015.

[82] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[83] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[84] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *CoRR*, abs/1702.05659, 2017. URL: http://arxiv.org/abs/1702.05659, arXiv:1702.05659.

[85] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL: http://arxiv.org/abs/1412.6980, arXiv:1412.6980.