Development of Components for Barrier-Free Pupillometry

A Dissertation Submitted for the Inauguration
for the Degree of

Doctor of Human Biology at the Faculty of Medicine of the Justus Liebig University Giessen
by

Ahmed Gdoura

Giessen, 2023

Justus Liebig University Giessen
Faculty of Medicine
Department of Ophtalmology

Examiner: Prof. Dr. Dr. Knut Stieger
Examiner: Prof. Dr. Dr. Vadász, István
Date of Doctoral Defense: 14 January 2025

Acknowledgments

I express my gratitude for the extensive and diverse support that I received in myriad ways throughout the journey of composing this dissertation.

Special thanks are due to my esteemed supervisors, Prof. Dr. Dr. Knut Stieger and Prof. Dr. Alexander Effland. Their generous sharing of expertise, personal support, and unwavering encouragement provided me with both guidance and energy, contributing significantly to the completion of this work.

I appreciate the freedom you granted me to explore my ideas, which enriched the depth and creativity of this research.

Furthermore, I extend my gratitude to the entire academic staff at the Ophthalmology Clinic in Giessen, with a special acknowledgment to Prof. Dr. Birgit Lorenz. Her instrumental role in facilitating my work on this deeply personal topic, coupled with her invaluable experience in refining the research focus, has been crucial to the success of this endeavor.

I would also like to thank my colleagues at the Optik Zentrum Wetzlar (OZW), with special appreciation for Prof. Dr. Markus Degünther. Our ongoing collaboration and his invaluable insights have significantly influenced and enriched the development of my work.

Lastly, my profound thanks go to my family and friends. Your boundless love, encouragement, and unwavering support have been the pillars that made these scholarly pursuits not only possible but also meaningful.

Contents

List of Acronyms

AAM active appearance model

ASM active shape model

BFP barrier-free pupillometry

BN batch normalization

CNN convolutional neural network

CRF Conditional Random Field

CTP Correct True Positive

DL deep learning

FCN Fully Convolutional Networks

ffERG full-field Electroretinography

FN false negative

FP false positive

GMM Gaussian Mixture Model

HOG Histogram of Oriented Gradient

IRD inherited retinal degenerations

ITP Incorrect True Positive

LCA Leber Congenital Amaurosis

ML machine learning

MRF markov random field

MSE Mean Square Error

NME Normalized Mean Error

PCK Percentage of Correct Key-points

PDM Deformable Parts Model

PLR pupillary light reflex

PWC pixel wise classification

TN true negative

TP true positive

–1–
Introduction of BFP Components

1 Pupillometry: The Classic Approach

Pupillometry is the field of study that involves measuring the changes in the diameter of the pupil: in response to a light stimulus, known as the pupillary light reflex (PLR), or the spontaneous variations that occur naturally in the pupil. It is performed in general by devices that are directly exposed to the patient's eye whether in a laying position such as tabletop devices or by standing or sitting such as for hand-held or table devices.

During pupillometry sessions, it is expected that patients have a minimum level of understanding of the technician's instructions. This includes cooperating by keeping their head pose and gaze fixed, refraining from blinking, maintaining a consistent distance from the device to ensure consistent image acquisition, and remaining calm and steady throughout the measurement period. Figure 1.1 illustrates some cases of traditional pupillometry sessions for adult patients.

Nevertheless, the aforementioned requirements could not easily apply to infants and

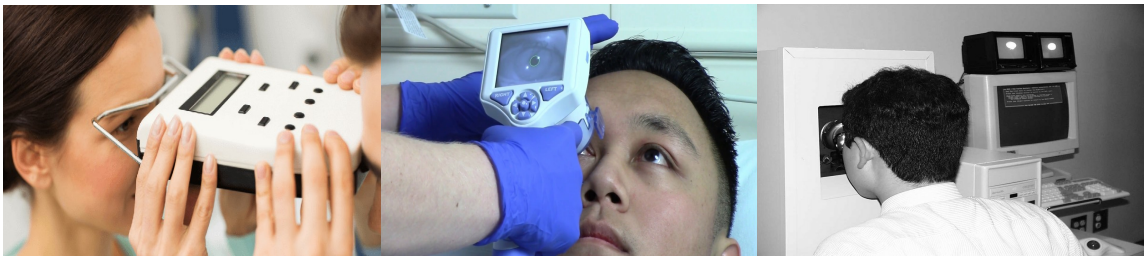


Figure 1.1: Classic Pupillometry Session: Indispensable Patient Cooperation[1]

young children. In many cases, pupillometry for such patients could provide valuable information about the therapeutic progression and the general state of the treated pathology [2].

[1] images from: https://www.kbvresearch.com/blog/pupillometer-help-treatment-neurological-disorders/ , https://youtu.be/EjlZ5oocl0g?t=31, and [1] respectively.

As a motivation for this work, a brief introduction about pathological cases that involve the exhibition of pupillometry for very young patients is presented.

2 Pupillometry as readout parameter in Retinal diseases

Inherited retinal degenerations (IRDs) are associated with mutations in more than 250 genes where some forms affect the retinal function via its photoreceptors from birth or early childhood. They are classified as Leber Congenital Amaurosis (LCA) or early childhood onset IRDs [3].

The extent to which rod pathway or cone pathway functions are affected varies with specific genes and disease duration. Quantification of the function of both, rods and cones, even when only residual, is therefore highly desirable, in particular as therapeutic options have recently become available and even approved for clinical use such as gene therapy as established by Russell et al. [4] and other pharmaceutical interventions. A standard method to quantify rod and cone function is full-field Electroretinography (ffERG). However, in severe and advanced forms of IRDs, ffERG is often not measurable. Therefore, alternative methods have been developed such as 2-color-threshold perimetry, Full-field Stimulus Test FST, and Chromatic Pupillometry [5]. Whereas 2-color-threshold perimetry and FST are subjective tests, pupillometry allows an objective measurement of rod and cone function [6].

Patients with RPE65-mutation-associated IRDs for example, suffer from severe night blindness from an early age and their functional vision is highly light-dependent [7]. However, special therapies significantly improved rod function and hence resulted in a much less dependent visual performance on high light levels [4]. Such therapies have been approved for patients from the age of 3 years on, but recently even younger children were treated (A. Nagiel, The Vision Center, Children's Hospital, personal communication).

Whereas subjective quantification of cone and rod function is challenging or even not feasible at such a young age, chromatic pupillometry, on the other hand, is an objective test that allows similar quantification of rod and cone function as chromatic FST [6]. However, the current measurement of chromatic pupillometry necessitates that the head and eyes are stable. Therefore, enabling chromatic pupillometry to be performed under these circumstances is the core of this work. Hence, developing a method to detect pupil position automatically and in real-time independent from the head and eye position is a necessary first step for barrier-free pupillometry (BFP).

Despite the continuous development in the field of pupillometry, especially since pupillometers are manufactured by several companies, the current products available on the market cannot meet the already-mentioned requirements of BFP. A brief overview of the different segments of development in pupillometry achieved by the main actors in the market is presented as follows.

3 Segments of Development in Pupillometry

Depending on their application, pupillometers on the market could split into tabletop or hand-held devices. Enhancements in both categories involve data accessibility, post-processing of the measurements, and integrating more features into the same device. The leading manufacturers that dominated the global market in 2021 in pupillometry were NeurOptics, Inc., Essilor Instruments USA, HAAG-STREIT GROUP, and SCHWIND eye-tech-solutions. Features integration enhancement concerns mainly the table-top category. For example, the LENSTAR device from HAAG-STREIT or the NPI-300 device from NeuroOptics could perform biometry, keratometry, pachymetry, and pupillometry measurements in addition to the IOL calculation in one click. Similarly, SWIND equipped their AMARIS 1050 excimer laser with a "7D" eye-tracking system able to perform very accurate pupil size estimation among several other parameters. Accessibility enhancement, however, is nowadays achieved by connecting devices to the internet and mobile devices such as the BRIGHT LAMP, Inc. company. Finally, data postprocessing enhancement involves pathology prediction such as neurological diseases. For example, NeuroOptics' devices exploit pupillometry measurement in predicting neurological decline following acute traumatic brain injury admissions. A graphic illustration of the three enhancement segments is provided in Figure 1.2.

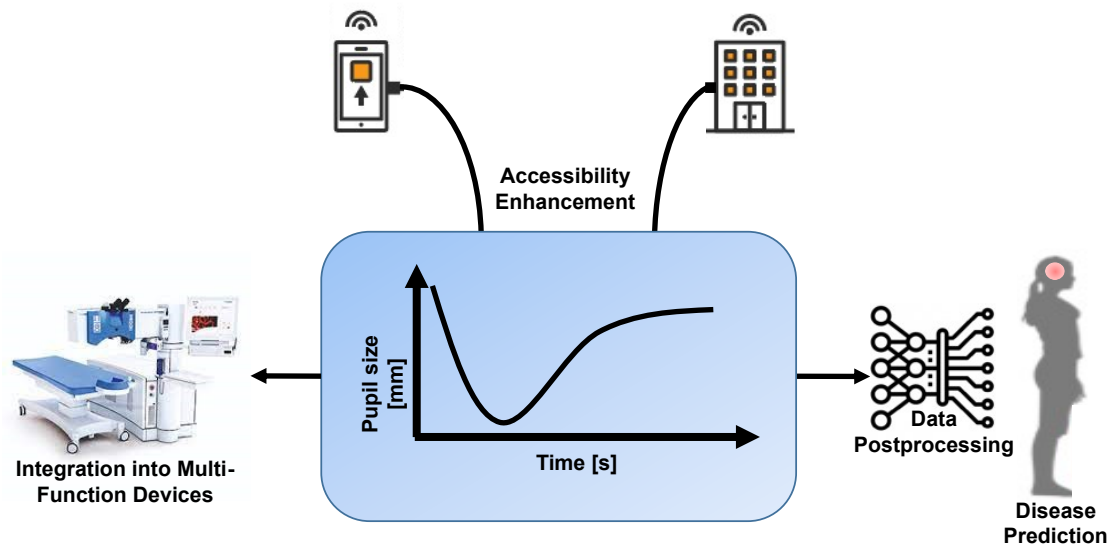


Figure 1.2: Enhancement Segements for Pupillometry: Accessibility, Integration, and Postprocessing

Based on the above-mentioned details, it is rather unlikely that new pupillometers can efficiently deal with the BFP circumstances in the near future. The absence of BFP-compliant products in the market can be attributed to the fact that potential users of these devices do not constitute a substantial market segment. As a result, the idea of producing BFP-conform products doesn't appeal to manufacturers because the potential

profits wouldn't justify the investment. Based on this fact, the main purpose of this work can be stated as will be detailed in Section 4.

4 Requirements for Barrier free Pupillometry

This section defines the primary problem addressed by this work. Specifically, the expected specifications from the BFP perspective are outlined. To translate these specifications into tangible features, the pupillometry processing steps are thoroughly examined and how each step relates to the BFP specifications is analyzed. By doing so, one can identify the key areas where improvements and adaptations are needed. Ultimately, the new features that will be developed for each processing step are introduced.

4.1 Problem Statement

During pupillometry sessions, young patients usually make spontaneous gestures such as changing their head pose and orientation, performing moderate body movement, modifying their gaze, blinking, or even conducting accommodation reflexes on external objects. The output frame characteristics, recording these circumstances, are hence totally different from those which are generated by eye-focused cameras equipping classic pupillometers. Therefore, new methodologies to address pupil size estimation should be imposed by this new image environment.

4.2 Purpose Statement

Formally speaking, the main purpose is adapting pupillometry to non-cooperative patients such as infants by supporting the different steps of the data processing workflow to incorporate a "barrier-free" functionality.
It is meant by a barrier-free pupilometer, (BFP), a pupilometer that tolerates small to moderate head poses, upper body movements, and gaze direction variations besides the original sources of variability affecting the traditional pupillometers such as blinking. Furthermore, BFP should not create a source of stress or discomfort for the patient since such factors highly affect PLR as proven in Beatty [8]. Meaning that BFP different components should neither be in direct contact nor too close to them. In other words, BFP noticeability must be retained as low as possible for an infant.
Before elaborating on the mentioned purpose, it is important to understand the main steps in performing pupillometry.

4.3 Pupillometry Processing Steps

Pupillometry is a composite process that involves different steps namely:

i) Stimulus presentation: where a specific stimulus is introduced to the subject which can be visual (such as the treated case of study), auditory, or cognitive.

ii) Pupil-size measurement: where the actual size of the pupil, generally ranging from $1 - 9$ millimeters, is extracted from an infrared camera frame. The different steps for pupil-size measurements will be detailed below as they present the main focus of this thesis.

iii) Data analysis: Once measurements are collected, the data is post-processed for denoising and filtering from artifacts and then analyzed using statistical means.

iv) Interpretation: This involves sanity checks of the measurements and examining pupil size changes according to the introduced stimulus. The interpretation can provide insights into the pathological state of the treated patient such as their retinal ability in absorbing and transmitting light via the different photoreceptors.

Elements ii) and iii) form together the data processing component in pupillometers which could be depicted by three main steps namely pupil detection, pupil size estimation, and sanity validation as illustrated in Figure 1.3. The pupillometry pipeline includes, in addition, data acquisition and data storage/export steps. These last two steps, however, do not present the focus of this work.

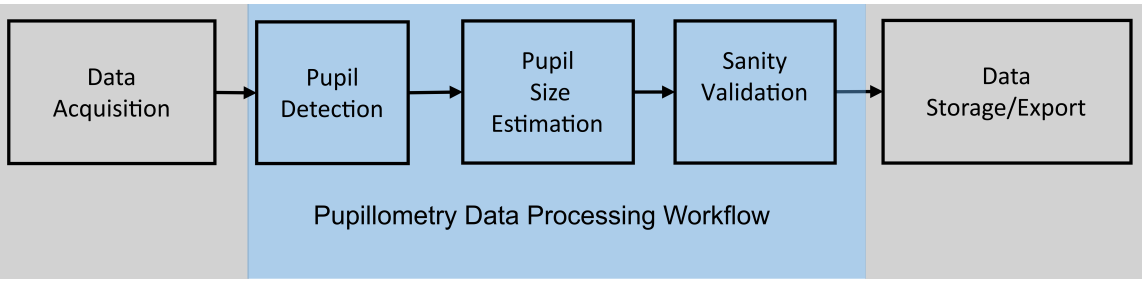


Figure 1.3: Classic Pupillometry Data Pipeline

It is important to note that the same data pipeline should remain unchanged even after integrating new BFP-conform features. Any algorithms developed to address these new features should be implemented solely under the predefined steps of the data processing workflow. Similarly, BFP should also start by detecting the pupil, estimating its size, and then validating this estimation according to the pre-set BFP specifications. However, despite adhering to the classic data workflow, BFP features have the potential to significantly reshape its components.

4.4 Effect of BFP Specifications on the Data Processing Workflow

The designated specification of the BFP concept should directly impact the acquired image space. More concretely, the condition of the free head pose together with the minimum level of noticeability will prohibit the exploitation of eye-focused images generated by narrow field-of-view cameras that usually equip classic pupilometers. Nevertheless, a wider field of view along which, a simultaneous recording of both “mobile” eyes is

required. Therefore, a new frame environment that contains the whole face preceded by a background should be expected. This must in turn drastically affect the pupil detection step since adapted tools are required to automatically extract the original pupil-centered ROI from the new image field.

Note that a successful extraction to the pupil-centered environment does not always guarantee a safe return to the classic pupillometry workflow. An allowed free-head pose will affect the distance of each pupil to the camera axis designated in the literature by the depth. This depth variation must, therefore, be considered at the pupil size estimation step for a reliable BFP. Depth variation, to the best of my knowledge, was never integrated into pupillometers for accuracy enhancement of pupil size estimation.

Another effect of BFP specifications on the pupil size estimation step manifests at the pupil edge segmentation level. As edges extracted from eye-centered input images should contain finer spatial resolution compared to the new frame space, an adapted pupil edge extraction is, therefore, the next manifestation of the impact of the BFP specification on the traditional pupillometry workflow. The expected method must be designed to overcome the degradation of the mm/pixel ratio.

Other challenges will also arise due to the new measurement condition that should not occur otherwise. For instance, pupil occlusion and out-of-frame pupils are two situations that could frequently occur under BFP circumstances and should be carefully addressed. These last two cases will require a general solution for the algorithm of the Sanity validation step. As for the classical devices, the only factor to examine was blinking, BFP however, should consider more sources of variability which turned the validation process into a more complex one.

Lastly, a novel feature under the data validation context is introduced, namely the Objectivity Enhancement of PLR. This functionality is a post-processing step that takes place after a successfully sanity-validated estimation. The purpose of this feature is to provide a decision support tool to recognize pupil accommodation which is a crucial factor that influences the PLR and therefore the collected pupil size.

Table 1.1 summarizes the set of specifications of BFP compared to the traditional pupillometers at different data processing steps.

Processing Step/ Data Characteristics	Classic Pupillometry	BFP
Input Image	Eye Centered	Face and Background
Pupil Edge Segmentation	Pixel Level	Sub Pixel Level
Size Estimation	Pupil Radius	Depth Consideration
Sanity Validation	Blinking Detection	Blinking, Out-of-frame, Occlusion
Objectivity Enhancement	-	Decision Support for Accommodation Recognition

Table 1.1: Classic Pupillometers Vs. BFP specifications

4.5 Aimed Features for BFP

Due to the new challenges for pupillometry faced in this work, new features were introduced for achieving reliable measurements. More concretely, a depth estimation module is integrated for the size estimation to deal with the free head pose condition varying the distance between the pupil and image acquisition system. It is worth noting that depth estimation is a well-established image-processing tool that was adopted via a well-known algorithm. In addition, pupil detection and pupil size estimation will be achieved via adapted methods to the new circumstances. Moreover, a decision support tool for objectivity estimation is introduced for dealing with the accommodation problem frequently emerging during the pupillometry sessions and interfering with the light stimulus by affecting the pupil reaction.

After having discovered the potential influence of the BFP specification over the classical pupillometry data processing components, one can in Section 5 explore the classic methods in dealing with them. Thereafter, the state-of-the-art methods that could eventually respond to those specifications are investigated in Section 6.

5 Data Processing for Classic Approaches

Whilst modern pupillometers could achieve high performance in terms of accuracy in pupil-size measurements, they are still confined to high-resolution eye-centered images.

A classical pupil size estimation process starts with frame acquisition followed by frame preprocessing for denoising and enhancement. Thereafter, pupil-region detection followed by pupil segmentation for size estimation steps is achieved to finalize the process. Whereas frame acquisition and preprocessing present well-established image-processing solutions, in the rest of this section, a brief review of pupil-region detection and pupil segmentation is provided.

5.1 Pupil Detection

The pupil-region detection step is heavily influenced by the treated frames' characteristics. As mentioned earlier, and based on the assumption of the absolute compliance of the patient, classic pupillometer recorded images are eye-centered ones. This section briefly presents the common methods, exploited by this type of pupillometers, in achieving this task.

Pupil detection algorithms from eye-centered frames usually assume that the pupil is the unique dark blob within a gray-scale image. This assumption transforms the pupil detection challenge into a blob detection one that can be addressed by several approaches in the image processing literature such as:

i) Gradient-based pupil center estimation determines the point where gradient vectors most intersect. This basic idea was adapted to better perform within the iris-pupil envi-

ronment by integrating prior knowledge about the eye appearance and gaining robustness against semi-circular shapes as proposed by Timm et al. [9]

ii) Haar-like features as exploited by Świrski et al. [10] to estimate the pupil region form an eye-centered one.

iii) Thresholding techniques rely on the statistical characteristics of the histogram to extract the appropriate threshold to segment the pupil region and get rid of the background. Such approaches are usually preceded by the morphological opening of the image to get rid of corneal reflections and partially the eyelashes to produce a more homogeneous pupil region easily segmented through the histogram.

5.2 Pupil Size Estimation

Open-source solutions for pupil segmentation (which also confounds with the pupil size estimation), could prove state-of-the-art performance in achieving very accurate pupil-size estimations (in the order of 0.01mm) at a very high fps (in the order of 120 fps). However, the suggested methods as in Zandi et al. [11], Santini et al. [12], or in Santini et al. [13], hypothesize eye-centered images over which the pupil detection algorithm is executed. A more in-depth overview of the structure of pupil-segmentation methods as well as a brief literature exposé about this subject is provided in Section 6.3.

6 Data Processing for BFP

After providing an overview of the methods adopted for various stages of data processing in classic pupillometry, this section focuses on different components that can address the specifications of the BFP. Specifically, state-of-the-art methods that address pupil detection tasks in this context are addressed in Section 6.2. Additionally, methods for estimating pupil size are discussed in Section 6.3. Furthermore, in Section 6.4, methods for detecting accommodation reflex are introduced. This serves as a means to filter out this factor from the collected signal, allowing us to obtain a light-dependent PLR rather than an accommodation-dependent one. It is important to highlight that open-source datasets for evaluating the performance of different methods are referenced.

This section starts by proving the urge to use a machine learning-based (ML) solution for detecting the pupil region as a preprocessing step for pupil size estimation as explained below:

6.1 The Urge of Using Machine Learning

The image context in BFP is different from the one in classic pupillometer frames. Usually, BFP generates face and background frames containing information about both pupils rather than eye-centered monocular frames. Therefore, the basic supposition about the pupil being the unique rounded dark blob is no longer valid within this new environment.

in Figure 1.4, it is clear that the nose holes and the shadow stains present very similar characteristics to the pupil. Assuming such a hypothesis is therefore no longer valid and detecting the pupil from a similar environment cannot be achieved unless adapted tools are exploited. Basic features such as pixels gray level or gradients are unable to

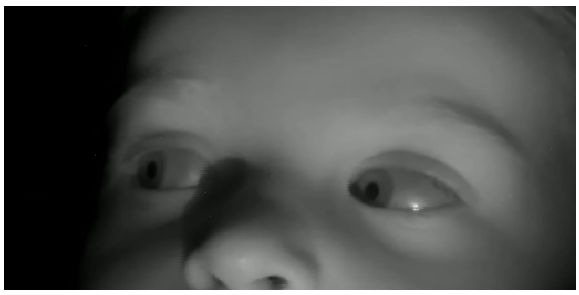


Figure 1.4: A manifestation of the non-validity of the pupil being the unique rounded dark blob hypothesis[2]

provide exclusively to the target pupils. In addition, pupil region segmentation based on such features usually runs iterative processes which makes them inappropriate for larger-size images catching a broader field of view. More complex features are therefore compulsory especially since such frames are more prone to be affected by different variability factors such as unconstrained illumination, extreme head-pose, exaggerated expressions, occlusions, and out-of-plane configurations observed on the treated data.

Since it is difficult to explicitly design features that can survive all the aforementioned circumstances, it is worth learning them via machine learning tools as will be presented in Section 1 of Chapter 2.

6.2 Pupil Detection for BFP

The problem of performing pupil detection in particular, as well as facial sections in general, under real-world conditions, remains unresolved. This category of problems is usually addressed in the machine learning field via facial landmarks detection means. Since its inception in the '90s by Takács et al. [14], achieving enhanced performance has consistently been a challenging issue. In the following, a bench of works addressing this challenge are listed under different categories.

6.2.1 ML-based Facial Landmarks Detection Categories

Attempts for Facial landmark detection could be split into three categories:

a) **Genrative Models**

Generative models in this context refer to the statistical models that estimate the facial landmarks' locations based on their probability distributions within the training datasets.

[2]image provided by the clinique of ophthalmology in Giessen

These distributions are referred to as shapes and build together a shape space. In the inference step, new unseen shapes are estimated within this space. Facial landmark estimation using this type of modeling can be addressed by two main categories as follows:

i) The Active Appearance models (AAM) introduced by Cootes et al. [15] and later adopted by Kopaczka et al. [16]

ii) The Active Shape Model (ASM) was introduced in [17] and later adopted by Hsu et al. [18].

b) **Discriminative Models**

Discriminative models in this context refer to feature-based models that estimate the facial landmarks locations based on inherently learned features from the training datasets. These features are learned through backpropagation algorithms. In the inference step, new unseen images are projected onto the new features space to output the new locations. Four different sub-categories can be spotted under this modeling style:

i) The direct regression as in Wu et al. [19].

ii) The heatmap-based CNNs as proposed by Merget et al. [20]

iii) The pixel-wise classification as in Khan et al. [21].

iv) the cascade shape regressions, where a set of regressors are learned to approximate the mapping between an initial shape and the ground truth as proposed by Deng et al. [22] and Liu et al. [23].

c) **Hybrid Algorithms**

The third category for facial landmarks detection is Hybrid algorithms which combine generative and discriminative approaches.

Medley et al. [24], for example, extract adequate features via CNN to learn the input for the optimization process of an ASM that finally detects landmarks.

In Deng et al. [22], the authors start with the Deformable Parts Model (PDM)-based face detection supported with a cascade shape-regression for incorporating a local refinement for the least accurate landmarks. Note that this regression relies on a multiscale Histogram of Oriented Gradient (HOG) features for estimating the landmarks.

In [25], a CNN combined with a Conditional Random Field (CRF) is jointly trained to predict a structured probabilistic estimation of landmark locations. This joint training succeeded in capturing shape deformation and pose variation simultaneously.

Finally, [26], originally designed for human pose estimation, conducted a joint training of a CNN and a Markov Random Field (MRF). When CNN performs an initial estimation of landmarks locations, MRF validates the pairwise relationships between them via the *Learned Conditional Distribution* of the location of one landmark relative to another. The degree of the spatial constraint of learned conditional distributions is a key factor for their

approach success. More concretely, the relative position of one landmark to its neighbor must be spatially consistent with its contextual limitations. Such limitations could be severely altered by back-facing frames or overlaps between body parts caused by extreme body poses.

6.2.2 ML-based Facial Landmarks Detection Challenges

Two major challenges face successful facial landmarks detection, i.e. the local context distinguishability, and global compatibility. While information about the first challenge could be acquired from the surrounding area of the subject landmark, solving the second one necessitates a way to enforce the model obeying the contextual spatial constraints dictated by the human face structure.

CNNs are powerful at distinguishing local features due to the relatively small size of the low-order kernels compared to the input image, however, high-order kernels lack the efficiency to learn global context due to the low resolution of the treated receptive fields.

As Yue-Hei Ng et al. [27] proved the quality of a feature gradually increases from the initial to intermediary layers to drop again when proceeding towards the final deep layers.

Generative models, on the other hand, can efficiently learn the higher-level constraints of the landmarks configurations. This permits enforcing the global spatial consistency on a given pre-estimated set of landmarks. In the following, a brief investigation of a bench of CNN-based attempts at facial landmark detection. Thereafter, introduce the choice concerning the generative model is introduced.

a) **Chased Characteristics for CNN**

This section reviews a set of CNN attempts to extract the aimed features that should equip the network.

i) Facial Bounding Box Free CNN

Relying on initial face detectors via bounding boxes estimation could be problematic for facial landmarks detection especially if the input image presents moderate to extreme head-pose or an out-of-frame part of the face. While Sun et al. [28] and later Chen et al. [29] relatively succeed in performing a cascade regression of facial landmark locations via a multilevel convolutional networks model, their prediction performance remains strongly dependent on the performance of the initial face detector and on the cropping process around the subject set of landmarks. Hence, it could be claimed that the chased model must be independent of any data preprocessing step.

ii) Ablation Tolerant CNN

Direct regression-based CNNs (for example, Sun et al. [28] and Zhang et al. [30]) where models seek to learn a direct mapping from the input image space to the landmark coordinates are unfit to deal with out-of-frame landmarks also called ablated frames. This type of regression initially presets the output dimensionality of the network, which makes them unable to deal with ablated frames that could contain a variable number of

existing landmarks. Moreover, this type of frame could not be integrated into the training process with these regression models since the optimization criterion requires fixed-length ground-truth coordinates of the landmarks. Furthermore, direct regression gives rise to a highly nonlinear mapping that is prone to poor performance. Heatmap-based CNNs present an alternative to alleviate the degree of complexity due to the proportionate similarity between input and output spaces. The performance of direct regression such as in [31] has been evaluated in [32, Table 9], where it was outperformed by all mentioned heatmap-based methods.

iii) Homogenous Architecture CNN

Terminating CNNs with a flatting layer followed by a set of fully connected layers is a common practice especially for the direct regression case to progressively down-sample the treated signal to meet the landmark dimensionality in their $2D$ coordinate space as in [28]. Fully Convolutional Networks (FCN) come to substitute such heterogeneous networks with special convolution layers to achieve end-to-end convolutional learning and inference as extensively argued by Springenberg et al. [33]. By avoiding the $1D$ flatting layer, FCNs could preserve the spatial structure within the image which alleviates the mapping complexity on the one hand, and avoid overfitting by requiring less number of parameters compared to the greedy signal flatting procedure on the other hand. Finally, FCNs architecture enables an arbitrary size for the input images a feature that cannot be achieved otherwise.

Note that Hannane et al. [34] could overcome the complexity of the learned mapping by a CNN-FC network for facial landmarks detection by sequentially reducing the input space via splitting regions around the treated landmarks. However, ablation could not be resolved as a fixed number of landmarks is expected.

iv) Heatmaps Based CNN

Heatmaps-based CNNs as a special type of FCNs are the adopted form for detecting facial landmarks in general and the patient's pupils in particular. This section presents a quick overview of this kind of CNNs which are considered the seed of this model. Long et al. [35] for example, transformed a classification network into a classification and segmentation task interpreted from an output heatmap. Other FCN categories such as the encoder-decoder style (such as U-Net introduced by Ronneberger et al. [36] and Hourglass networks presented by Newell et al. [37]) and the decoder-only network's style also known as heatmap regression networks as introduced by Bulat et al. [38]. The pixel-wise classification (PWC), was achieved via the encoder-decoder FCNs, originally introduced by Long et al. [35] by generating heatmaps with the same size as the input. The decoder-only network generally produces a lower-size heatmap that is interpreted as a probabilistic indicator of the locations of the facial landmarks.

based on the facts above, the heatmap regression is adopted for the challenge. Compared to the direct one, this type of regression commonly requires a relatively low number of trainable parameters due to the high similarity between the input and output image dimensionality. In addition, unlike direct regression, heatmap regression, due to its flexibility toward input and output dimensionality, can naturally handle amputated frames;

a source of variability that should be quite expected in real-life data and free-head-pose tracking experiments.

b) **Post-processing by Generative Models**

The usefulness of Generative models for deep learning has been studied by Erhan et al. [39]. They could be exploited to refine the CNN initial estimations by improving the global spatial consistency of their final output. These spatial consistency constraints could be expressed via information about the facial landmarks' interconnectivity easily incorporated into the learning process. Graphical models in general and MRF or CRF, in particular, are popular options to integrate landmarks interconnectivity. They both model geometric properties such as shape, spatial relationship, and connectivity among landmarks by estimating conditional probabilities of one landmark given the locations of the rest of the predicted ones. The implicit conditional probabilities learned from the training data can be exploited for a sanity check of the spatial consistency estimations by measuring their degree of conformity with the estimated conditional probabilities. Furthermore, whether by adopting an approximation as in [25] or integrating the exact formulation of the probabilistic graphical model as in [26], the main challenge remains their combination with the CNN to build an end-to-end system for learning and inference.

6.3 Pupil Size Estimation for BFP

Pupil segmentation also confounds with pupil size estimation, is a process that exhibits different steps:

6.3.1 Pupil Edge Segmentation

Ellipse detection algorithms generally and pupil edge segmentation more particularly rely on edge features in images for achieving their tasks. Three main sections can summarize such algorithms:

i) Edge Preprocessing:

Where detected edges undergo a bench of operations for elimination or enhancement based on curvature estimation, smoothing, and reduction techniques.

ii) Edge Matching

The remaining enhanced edges undergo a classification technique to allow their matching for incrementally building ellipse branches used for ellipse parameters' identification.

iii) ellipse candidates' searching

Ellipse branches, built over matched edges, could represent different ellipse candidates. This step aims to find a valid candidate among them and exclude false positives. This ellipse validation relies on sanity scores calculated for every candidate helping their comparison.

Even though these basic steps could summarize the ellipse detection algorithm, they were tackled in the literature by different approaches and organized on different workflows. In the following, the reader will discover different attempts for each step mentioned above:

a) **Edge Preprocessing**

Edge smoothness is a deciding criterion for their eligibility to be integrated into an ellipse branch. The concept of smoothness, due to the quantification error, is only roughly defined. Nevertheless, one should gain a clear understanding of it by analyzing a set of parameters and applying a set of measures over edges to enhance their smoothness as detailed below.

It is common to start the smoothness enhancement procedure with a length investigation of edges. The minimum edge length is an indicator of their smoothness since too short edges fail to show enough curvature of the ellipse or to run the parameter identification process that requires a minimum of five pixels.

Junctions omission also called orthogonal connections, is compulsory for edge smoothness enhancement. Edges endpoints, on the other hand, could contain valuable information for investigating their smoothness.

Curvature variation is valuable information for clearly disclosing the smoothness state of an edge. However, due to the noise and the quantization effects, it is challenging to exploit this criterion as it relies on tangent parameters estimation poorly defined under such effects. Note that different approximation approaches were adopted for edge curvature estimation under such artifacts.

Curvature estimation usually enhances edge smoothness by detecting and omitting irregularities such as sharp turns and inflection points. The omitting procedure will split the edge to generate a subset of smooth sub-edges more eligible to be merged into an ellipse branch.

In the following, a bench of algorithms for edge smoothness enhancement are consulted:

In [40], edge smoothness for ellipse detection is enhanced by junction removal following the code provided by Kovesi [41]. Thereafter, the edge curvature is approximated via the set of lines segment transform following Prasad et al. [42]. The amount of change of curvature is considered to detect inflection points and sharp turns and therefore split the subject edge at this location.

In [43], they start by filling the gaps between the too-close edges via linear interpolation thereafter they adopt a computationally efficient curvature approximation algorithm to spot irregularities in detected edges.

For the edge enhancement algorithm, Santini et al. [12] applied to the pupil detection context, is commenced by an edge thinning and straitening procedure based on hand-crafted kernels applied to the detected edges followed by breaking up edges at the detected junctions locations. The resulting edges are individually approximated by a set of dominant points following the k-cosine chain approximation proposed in [44]. Edges exclusion

is then performed by checking the dominant points' cardinality to omit those who are under 5 points. The next exclusion criterion exploits what they call edge diameter relative to a predefined range of tolerance built upon human pupil characteristics. Note that an edge diameter is defined as the maximum distance between two points within this edge, a computationally expensive process. By reaching this level, the curvature is approximated for the remaining edges. The curvature is calculated based on the ratio of the containing rectangle sides and too-straight edges (too-small ratio) are then discarded.

In [10], even though they followed a relatively different flowchart of pupil outline detection. The Edge enhancement is first performed only after a succession of ROI-diminishing procedures via the Starburst algorithm in [45].

b) **Edge Matching**

After having transformed the detected edges into smooth ones, ellipse branches must be built. This process usually starts with edge classification or a definition of an edge-specific search region within which another edge could be merged into the subject one.

In [40] search regions for every candidate edge are built based on the tangents of their endpoints and their middle points. The inclusion and exclusion decisions were then met based on predefined criteria.

In [43], edge matching is decided based on convexity and search region constraints. The mutual convexity constraints are fully defined based on the relative direction between the lines joining their endpoints at the midpoint level. On the other hand, the region constraints are defined based on the tangents at the endpoint level. Notice that these tangents were calculated based on the last n point at each edge side.

In [12], candidates' edge matching is performed via the relative position of their bounding boxes. The new edge is admitted to the candidates' edges list if and only if both pass a validation process and enhance the confidence measure of its components.

In [10], the matching process was integrated with the ellipse candidate searching one. They exploited RANSAC as an inclusion and exclusion procedure for the remaining edge pixels.

c) **Ellipse Candidates' Generation**

Reaching this level suggests that the dealt-with edges space has been judiciously reduced. This reduction as already illustrated was achieved by lowering the edges' dimensionality by introducing representative points instead of considering all the pixels within the edge, then excluding the too-short and too-straight ones. Thereafter, space reduction was carried out by merging edges to construct ellipse branches. At this level, an optimization algorithm is in general addressed.

Since Świrski et al. [10] did not apply any edge exclusion criteria they exploited RANSAC to filter false positives from the parameter identification process of their ellipse.

In [12] however, they investigate the contrast between the inner and the outer regions of an outline based on the contrast assumption of the pupil relative to the IRIS. Note that this

image awareness is also adopted by [10] by considering the degree of agreement between the under-investigation ellipse points gradient and the image gradient all integrated into a function called "support function".

Wang et al. [43] adopted a saliency score to evaluate different aspects of the geometrical quality of the ellipse uniformly and jointly. Three different techniques for the ellipses saliency scores are investigated in the following.

d) **Ellipse Saliency Techniques**

The ellipse saliency in [40], is based on three criteria i.e., the angular circumference ratio, the alignment ratio, and the angular continuity ratio. The first ratio expresses the angular range that covers an elliptic branch relative to its center. The second criterion is the ratio of the sufficiently close pixels to the ellipse hypothesis over the total number of the involved pixels. The third criterion expresses the angular continuity of the supporting edges of the ellipse hypothesis. The degree of angular continuity between two edges is measured by calculating the angle between them based on their tangents at their endpoints. Note that the first and the second criteria are less affected by the quantization error, unlike the tangent-based criteria which are drastically affected by this error.

Saliency in [43], was validated via four criteria as explained below: The shape and size expressed via their major and minor axis compared to preset thresholds. The fitting error expresses the closeness of the fitted pixels to the ellipse hypothesis. In the third criterion, the number of the involved edge pixels is compared to the ellipse perimeter. The last one investigates the angular coverage similar to the first criterion in Prasad et al. [40].

In [12], three saliency parameters were adopted: The ellipse aspect ratio between the minor and major axis. The angular edge spread is roughly approximated by the ratio of edge pixels in different centered quadrants of the ellipse hypothesis. The third criterion investigates the ellipse outline contrast which is estimated by comparing the intensity of the pixels in the inner and outer regions of the ellipse outline to validate the pupil appearance hypothesis (a darker region surrounded by a brighter region).

6.3.2 Depth Integration

Pupil size estimation is usually confounded with the pupil radius estimation and expressed in mm. While radius information extraction from ellipse fitting results is straightforward, a correction of this parameter by a depth-specific scale is integrated to integrate the depth variation factor and therefore enhance the overall accuracy of the pupil-size estimations. Recently monocular-based methods for depth estimation experienced a real improvement due to their reliance on ML-based algorithms. Ming et al. [46] illustrated the time evolution of monocular approaches for depth estimation and argued that Deep Learning (DL) models can provide high accuracy and real-time solutions.

Nevertheless, BFP requires a large field of view to integrate the free body and head-pose specifications, hence, the binocular (also called stereo) option for depth estimation is exploited. This well-established approach processes simultaneously recorded two frames

to extract the depth information. However, reliable results aren't guaranteed unless geometrical and contextual constraints are respected. The geometrical requirement arises from the relative position constraints of the exploited cameras necessary for the so-called camera calibration procedure. In addition, stereo-view algorithms rely on a disparity map as an indicator for the depth parameter. Building this map exhibits, in turn, a stereo correspondence procedure that relies on the contextual characteristics of the treated image and usually requires distinguishable features.

For illustrative purposes, the main components of the stereo-view-based depth estimation process are introduced:

i) Intrinsic and extrinsic camera calibration

ii) Stereo rectification

iii) Disparity calculation

iv) Depth information extraction

Note that these steps will be adapted to the BFP specifications to avoid any type of computational redundancy. Finally, This setup will enable extracting real-life pupil sizes instead of pixel-based ones. The adopted depth estimation Algorithm is presented in detail in Chapter 2.

6.4 PLR Objectivity

The main goal behind BFP is to measure accurately and objectively PLR under the chromatic light stimulus. While accuracy challenges are addressed in all the processing steps, this section is dedicated to PLR objectivity enhancement. In this context, the term 'objectivity' is used to refer to the fact that PLR could be generated by factors other than the light stimulus. In this case, a PLR presenting a combination of these factors is usually confounded with a stimulus-dependent signal. Dissociating these factors' contribution from the resulting signal is challenging unless careful measures are taken.

6.4.1 Objectivity Affecting Factors

Several factors can influence PLR objectivity on different scales such as:

Cognitive Stimulus: cognitive process results in pupillary dilatation. As it was illustrated in [8], the magnitude of this dilatation is relative to the cognitive load, short-term memory, language processing, Sustained attention, reasoning, and visual and acoustic perception. All but the last two factors are irrelevant to the experiment, as the infant patient will not be required to perform any mental effort. However, the visual and acoustic perception factors should be minimized. For example, acoustic perception, especially during low illumination light stimulus, should be avoided as the resulting dilatation is in the same order of magnitude as the PLR (0.1 to 0.2 mm Beatty [8]).

Visual perceptions, however, could be minimized by adopting adequate tools for the setup building such as adopting: low contrast structures with a smooth texture, a large enough light screen stimulus to avoid their edges perception, recording cameras with small apertures and with the same color as the screen

Emotional Influence: Henderson et al. [47] showed that PLR toward images with the same illumination levels differs depending on the psychological impact of the image content. Higher pupil dilatation was proved when showing pleasant or unpleasant scenes compared to neutral ones. To address this factor, the technician must ensure the patient's calmness, by easing their anxiety letting parent members accompany them during the measuring session, and letting enough time before launching the procedure to allow familiarization and adaption to the surroundings and the presence of the individuals.

Accommodation: the presence of visual stimuli influences the PLR. However, their presence could also launch more affecting PLR factor, which is accommodation. The accommodation reflex was described by Fisch [48, Page 208] as a three-part process: lens thickening, pupillary constriction, and inward rotation of the eyes—eye convergence. On the other hand, the order of magnitude of accommodation is comparable with PLR even in high illumination levels (Kasthurirangan et al. [49, Figure 5, 6] and Yu et al. [50, Figure 2, 3]). The adopted Strategy against this factor is to detect it and apply a correction factor to the PLR signal as a filtering procedure for accommodation. In the worst-case scenario, measurement, during accommodation, must be discarded.

Mind wandering: also called Shift of focus, is a pupil-dilating factor (0. 1mm [51]). The technician must signal and intervene to alert without causing emotional discomfort for the patient. The collected PLR from the Mind Wandering labeled frames label must be compensated.

6.4.2 Accomodation Investigation

Accommodation is believed to be the most challenging factor affecting the patient's PLR for three reasons: First, it affects the pupil reflex with the same order of magnitude as PLR does, second, it is very likely to happen, and third, it is hard to be noticed by the technician to interrupt it. To the best of my knowledge, Accommodation detection was not yet addressed to enhance the PLR signal objectivity.

Three different actions can reveal the occurrence of the accommodation process: eye convergence, pupillary constriction, and lens accommodation. This can be addressed by measuring the eyeball convergence, investigating the pupil constriction, or directly measuring the dynamic accommodation using auto-refractors. Even though the last parameter presents a direct quantification of the accommodation state, unfortunately, it suggests the use of external tools (auto-refractor), which is in contradiction with the adopted barrier-free philosophy and thus cannot be exploited. Ocular convergence and pupil construction parameters, on the other hand, could be determined computationally. Furthermore, it is required that these tasks must rely on the main processing stream for BFP. More clearly, the main module for pupil size estimation must also be used for accommodation detection.

Since BFP continuously checks pupil constriction, it could be revealed that the lacking feature for this task is the eyeball convergence usually conducted via gaze estimation.

6.4.3 Gaze Estimation

Gaze direction also referred to as the direction of the point of regard, was addressed by several approaches over the last decades. Gaze estimation methods are categorized based on two main criteria.

i) The intrusiveness: which is determined by the adopted sensing method that can be whether intrusive (electrodes, coil of wire, Active lighting) or non-intrusive (one-camera or multi-camera setups).

ii) The adopted method: gaze estimation methods fall generally into direct measurement, model-based, appearance-based, or hybrid techniques. Whereas direct measurement usually applies to the intrusive variation of the gaze estimation, the rest of the methods exploit the generated images and videos to fulfill their tasks.

It is worth noting that combinations and sub-categories among these methods can be derived. For example, the model-based category can be performed by an explicit model fitting or by model interpolation. On the other hand, appearance-based approaches could be addressed by conventional hand-crafted features or learned features usually leveraging DL-based tools.

The following is a brief illustration of the progression of gaze estimation methods in dealing with the faced challenges.

Early attempts for gaze estimation relied on IR light enabling glint-pupil vector acquirement as a direct indicator for the gaze direction. Even though active lighting solutions continue to be the reference in terms of precision as stated in Shin et al. [52], they still suffer from different issues. The glin-pupil vector calculation relies on a robust extraction of the glint from the eye image usually requiring specialized lighting properties varying with the recording environment. In addition, to decrease the depth-dependent parallax issue [53], such lighting techniques are usually combined with head-mounted setups or near-to-head camera setups which increase their level of intrusiveness and drastically limit the application space. An accurate glint position estimation is not possible if the scene plan lays beyond the tolerated range of depth, as the variation of the glint position will decrease with depth as stated by Hansen et al. [54]. To overcome those complications, gaze estimation techniques developed into less intrusive and independent processes based on explicit modeling of the eye including the $2D$ regression and the $3D$ modeling approaches. However, model-based approaches were restricted to data with limited scenarios and could not overcome the appearance, illumination, head-pose, and subject-specific variances. Later on, appearance-based methods including hand-crafted or learned features were adopted to address these variances. More concretely, appearance-based methods do not rely on geometrical and explicit features of the eye, however, eye imagelts are used to build a mapping to the gaze direction space.

In the following, a brief overview of appearance-based approaches for estimating the gaze direction is provided.

6.4.4 Appearance Based Gaze Estimation

Appearance-based studies for gaze estimation were tied to liberating it from geometrical models and pointed toward extracting this information directly from the image. They could be split into conventional and DL-based ones leveraging hand-crafted or learned features respectively. Even though the first category usually uses lightweight models and achieves relatively accurate results compared to the explicitly modeled ones, they usually fail to treat unconstrained data as they rely on fixed to limited head-pose and subject-specific data.

By examining the reviewed literature in the appearance-based section in Cazzato et al. [55, Section 7.2], It is clear that as time goes on, features derived from eye images are gradually replacing those that were handcrafted. This is due to the belief that the provided eye images should implicitly integrate other key features decisive for estimating the gaze direction, in addition to the geometrical parameters.

In addition, as studies become less restrictive toward the exploited datasets, head-pose variance presented a real challenge for appearance-based methods exploiting eye images. The poor performance was argued with the similar eye image appearances coming from different head-poses and gazes.

Sugano et al. [56] attempted to address the variant head-pose configurations by clustering the different head-poses based on a closeness criterion and interpolating the k-nearest neighbor labels within the subject's cluster to estimate the gaze. Their model input is the concatenation of a cropped eye region-based vector x and the estimated head pose p, $f = (x, p)$. However, they could only estimate an approximate region where the user is looking and achieved only around 5 degrees of gaze accuracy. They justified their performance to the adopted distance function measuring the closeness between samples as they argued its sensitivity to the eye cropping generated shift procedure.

A more recent alternative to address this problem was by explicitly providing or separately estimating head-pose information via an annex convolutional streams from full-face images as suggested by Zhang et al. [57] and Krafka et al. [58] respectively. Even though such techniques showed some improvement in dealing with head-pose variances, they are still exploiting limited ranges of freedom of this parameter (for example ±30 degrees and only 40 degrees around the yaw and pitch axes in [58]).

The second challenge appearance-based approaches are facing is subject-dependent variances. To address the cross-subject limitation, Mora et al. [59] trained their model on a multi-subject dataset integrating free head-poses configurations. Even though their experiments were restricted to only 5 subjects, they could not achieve less than 10 degrees of accuracy with their generic model and with a large amount of labeled data frames. Their model is limited to the linear combination of the 5 users' built templates. In their attempt to overcome the illumination changes, they built descriptors out of the transformed eye images by explicitly shifting the mean and the standard deviations of the gray levels and

by subsampling the resulting image into a 3by5 grid. This could turn to the fact that they regress unseen appearances by interpolating the appearances of the trained ones.

In [57], authors clearly stated that any proposed solution, for everyday scenarios, has to be person-independent, intrusion-free, and head-pose invariant. Their suggestion was to combine the model-based gaze estimation with appearance-based gaze estimation challenges to overcome their challenges. They argued this with the fact that appearance-based should have access to good labeling quality data that is diversified in terms of illumination and head-pose, model-based approaches, on the other hand, must exploit models that have the potential of building the regression mapping.

Later, studies tend to rely less on model-based approaches for the benefit of the DL appearance-based ones. Park et al. [60] exploit Hourglass CNN as a DL model to improve head-pose invariance without requiring any other external information. They built an intermediate representation for their model learning called the gaze map. This *"abstract, pictorial representation of the eyeball, the iris and the pupil at its center"* was built from eye images and their corresponding gaze direction labelings and later exploited as an intermediate station for the eye-image to gaze mapping as a procedure to relax its complexity. A considerable improvement in gaze estimation performance in unconstrained data was reported.

6.4.5 Data Sets

Appearance-based gaze estimation, as a deep learning-based solution, relies substantially on the provided datasets. It is recommended to refer to Cheng et al. [61, Table IV] for an updated illustration of gaze estimation datasets. The authors refer to the provided annotation for every dataset. In the following, a set of published datasets for gaze estimation is presented.

In [62], the authors built the dataset by showing a white circle on a black background with a red cross at the center. The monitor was split onto a 16 by 10 grid. First, the circle appears together with its label at the center, later, the circle vanishes while the red cross remains. At this moment, the cameras are triggered to record. As the author postulated, the data set provided the gaze directions for approximately ± 25 degrees horizontally and ± 15 degrees vertically, which should cover the range of natural gaze directions. The question that could manifest here is whether this practice is biased by the accommodation that can be started on the circle center as a fixation point.

In [57], the authors exploited the same shrinking circle technique for gaze collection where participants were asked to fixate on the marked center of this circle.

–2–
Methods for BFP Components

This chapter introduces the adopted solutions for addressing the aforementioned BFP components as illustrated in the Chapter 1 of this thesis. Three main packages are suggested for solving the BFP challenges. First, an end-to-end CNN for pupil detection from face and background images is presented in Section 1. Secondly, a novel approach for pupil edge segmentation which is annexed by depth-based pupil-size correction is presented in Section 2. Thirdly, Section 3 presents an algorithm for investigating accommodation that utilizes gaze direction estimation and the evolution of pupil size to enhance the objectivity of PLR."

1 Pupil Detection

This section presents the proposed algorithm for detecting pupil region form images under free-head pose conditions as a preprocessing step for pupil-size estimations.

The proposed method for facial landmark detection exhibits two major components: the landmark detector (see Section 1.1) that estimates the location of the landmarks based on heatmaps representing the probability of their occurrence at a certain position. In a post-processing step, the spatial model (see Section 1.2) verifies the pose consistency of a landmark relative to the other ones following a Markov Random Field-like graph. Finally, the adopted loss function that enables simultaneous training of both model's components in takes place Section 1.3.

As studied by [63], a useful feature map is a representation that includes

i) high-level features generated from a sufficiently deep network to encode high-level object knowledge,

ii) fine spatial details around the object to learn its discriminativeness, and

iii) an explicit internal representation of entities and their relationship to associate components with one another.

The hybrid model should therefore handle the aforementioned requirement through the learned features. More concretely, the landmark detector is designed to handle Item i) and Item ii), whereas the spatial model is responsible for Item iii).

1.1 Landmark Detector

This section introduces the adopted CNN-based landmark detector, which is designed to generate heatmaps (one for each predefined landmark) reflecting the probability distribution of a specific landmark being located at a specific position.

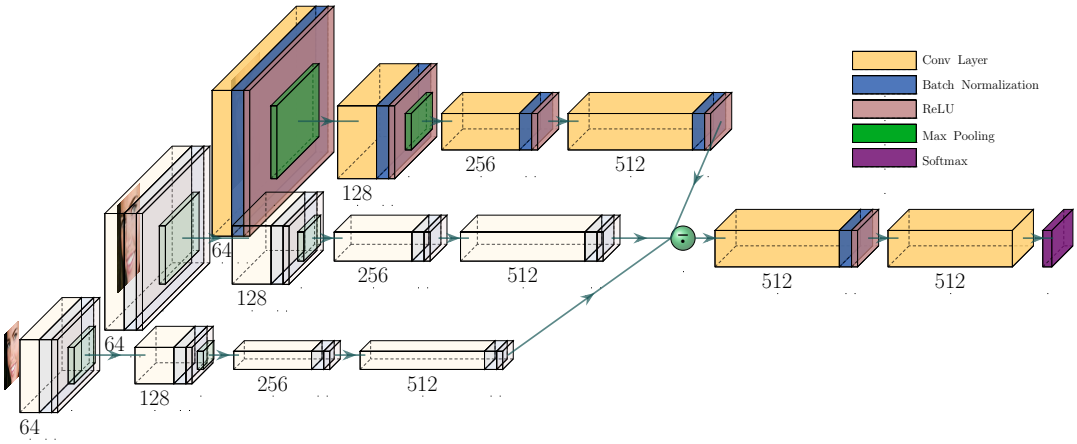


Figure 2.1: The proposed CNN Network: transparent effect means no new layer, the average is computed at the green circle level. (Best viewed in color.)

The network depicted in Figure 2.1 exhibits two subparts S_1 and S_2:

(S1) Each image is processed on three different scales by four consecutive convolutional blocks, which essentially extract low-order features.

(S2) Subsequently, the average of the results of the previous subpart represents the input for the remaining convolution layers, which extract the higher-order features. to ultimately generate the *LandmarkDetector*'s output.

This architecture, via its two subparts, is designed to equip the LandmarkDetector with discriminative local features simultaneously with enough high-level knowledge as explained in Section 1.1.2. The distributed design, on the other hand, tends to explicitly learn scale-invariance to efficiently deal with variations in the sizes of the objects as explained in the forthcoming subsection.

1.1.1 Scale Variance Handling

Following [64], scale invariance cannot be considered as an intrinsic feature for CNNs, which is additionally strongly affected by the image resolution. Therefore, scale-invariant

assumptions about scale-variant information must be explicitly learned if such a feature is pursued, which in turn can be achieved via a direct exposition of the model to this type of information at different scales.

The straightforward method is augmenting the training data by applying scale jittering. This approach could be seen as a technique that zooms partially and arbitrarily images within a predefined scale range. Models typically show a lower test error as they are exposed to more data that in principle, should reduce overfitting. Though this might be satisfying at first glance, [65] showed that this technique will push the model to require more scale-variant versions of the same learned feature instead of learning one scale-invariant feature. Unfortunately, this increases the model size especially when many scale levels are introduced, and results in overfitting.

The second approach to deal with scale variance is by adopting separate CNNs training at different scales. The final output is then the average of their estimations. It is worth noting that his approach suffers from redundancy, in particular for scale-invariant or high-level features. In addition, it cannot scale up if a large scale range is expected. Therefore, the Item (S1) is designed to systematically incorporate each data point at three different scales through the same layers to overcome redundancy.

In other words, exposing the model to each data point at different scales will not require introducing new convolutional layers. This technique aims to push the first convolutional blocks of the network to build scale-invariant representations out of scale variant features. Hereafter, the average of all processed scales is fed to the subpart (S2) of the LandmarkDetector to generate the final heatmap distribution. It is worth noting that image sub-sampling for generating the different scales should not allow spatial spectrum distortion also known as aliasing. This risk increases with the down-sampling order as the sampling frequency will decrease and will then violate the Nyquist-Shannon principle. Such a violation manifests by an overlapping of the high-frequency components of the image's spectrum, and by high-frequency artifacts in the image space. To avoid this phenomenon, low-pass filtering is applied before every down-sampling step.

1.1.2 Low-/High-order Feature Compromise

Contrary to local features, global features allow for a more generalized interpretation of objects due to their larger receptive field. However, this extension of the receptive fields (generally by applying size reduction layers in the CNN) deteriorates the spatial resolution due to the application of several sub-sampling steps and therefore fails to provide local context variation. In particular, very deep networks are inferior when dealing with local distinguishability. Inspired by Kim et al. [66], the network is consequently widened to promote the depth of the learned feature and the distinction of the local context simultaneously.

Consequently, motivated by the original target of building a fine-spatial, sufficiently deep, multi-scale handling feature map, a CNN of a distributed architecture is adopted to run the Gaussian pyramid of every image as illustrated in Figure 2.1. It is worth noting that the shallow scale-distributed architecture of the LandmarkDetector's subpart (S1)

preserves spatial affinity and therefore enables the pursued local discriminativeness. As for the high-level features, the network's second subpart (S2) is equipped with high-dimensional convolutions which are essential for their learning.

Finally, the generated heatmaps present landmark-specific unary distributions revealing the probability of the presence of the subject landmark at each pixel's coordinates. Thereafter, these heatmaps are fed to the SpatialModel to perform a spatial consistency check of each detected landmark relative to a predefined set of other landmarks as detailed in Section 1.2.

1.2 Spatial Model

Given the initial landmark's unary distributions estimated by LandmarkDetector, and revealing their locations, an MRF-like post-processing is launched to validate their relative spatial consistency.

For this purpose, every landmark i is treated via a landmark-specific graph model G_i, built out over a predefined set of its neighbors. Thereafter, the MRF-like process is run over the vertices of G_i to catch spatially correlated features characterizing their mutual influences. In this process, the learned conditional distributions are extracted from the training data as presented by Jain et al. [67] and detailed in the next section:

1.2.1 Learned Conditional Distribution

The learned conditional distribution for a pair of landmarks (i, $cond_i$) noted $p_{i|cond_i}$ is determined offline before starting any model training procedures. For each image, the landmark i is translated with the same amount that would shift its conditional landmark $cond_i$ located at (u_{cond_i}, v_{cond_i}) to the frame center. This translation amount is quantified by:

$$T_{i|cond_i} = \text{center of frame} - (u_{cond_i}, v_{cond_i}) \tag{2.1}$$

After landmark coordinates were transformed, conditional probabilities were built for every $i|cond_i$ combination by fitting the collected data to a Gaussian Mixture Model (GMM). GMM assumes that all the data points are generated from a mixture of a finite number of Gaussian distributions, which is also called the order, with unknown parameters. Order determination was achieved via an exhaustive investigation of the fitting performance based on predefined scores. The best GMM order has been set as the average of the minimum of the well-known AIC and the BIC statistical scores. Note that the resolution of the learned conditional probabilities is twice the resolution of the estimated heatmaps. Figure 2.2 illustrates the conditional probability $p_{lmouth|nose}$ revealing the locations of occurrence of the left extremity of the mouth when the nose tip is located on the frame center $center_{120x180}$.

In [67], the pre-estimated heatmap of a landmark has been filtered by the learned conditional distributions of its direct neighbor according to an approach analogous to the sum-product belief propagation algorithm introduced by Felzenszwalb et al. [68]. In the

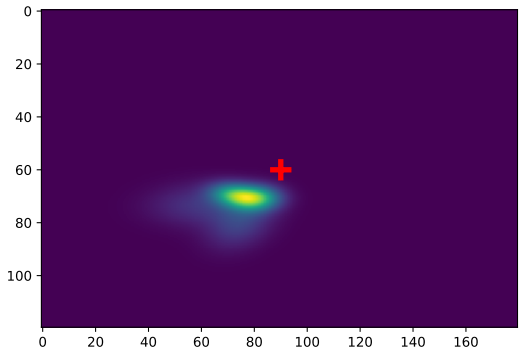


Figure 2.2: :$p_{lmouth|nose}$ the left mouth extremity spatial distribution when nose landmark occupies the heatmap center (+) (Best viewed in color.)

treated case, a broader neighborhood space is explored to validate the spatial consistency of the subject landmark.

1.2.2 Neighborhood Space Definition

To define neighborhood systems for the MRF as a neighborhood-based graph model, it proceeded as follows: Let $S = \{1, 2, ..., n\}$ be the set of n landmarks and $i \in S$ a specific landmark. Then the associated local neighborhood $N_i(r)$ for the landmark i given a radius $r > 0$ reads as:

$$N_i(r) = i' \in S : dist(i, i') \leq r, i' \neq i \tag{2.2}$$

where $dist(i, i')$ is the Euclidean distance between i and i'. In addition to the local neighborhood, the fixed set of global reference landmarks $N_g \subset S$ is considered, which encompass distinct particularly conspicuous facial landmarks.

Finally, the customized neighbourhood $N_u(i)$ is the union of $N_i(r)$ with N_g such: $N_u(i) = N_i(r) \cup N_g$ as detailed in Figure 2.3.[1]

It is worth noting that the choice of conditional landmarks obeys the following rules:

- $N_i(r)$ presents the local consistency challenge that prioritizes the nearest neighbors over farther ones and provides an image of the local state around the subject landmark.

- N_g presents the global structure of the human face that prioritizes some landmarks, which are called central landmarks, over others. N_g roughly indicates the smallest set that describes most of the face structure.

[1]Image by Vincent Angler (CC BY-2.0), `https://commons.wikimedia.org/wiki/File:Croydon_facelift_2012.jpg`

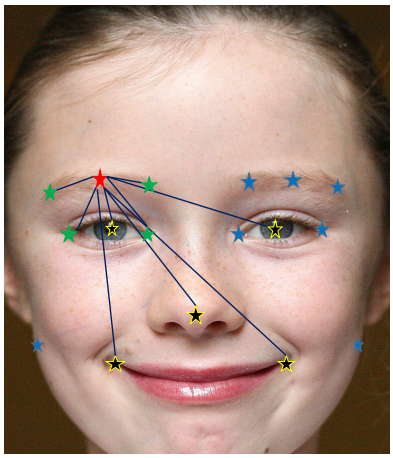


Figure 2.3: MRF Graph G_i for a specific landmark (red) linked to its local neighborhood $N_i(r)$ (green) and the global one N_g (yellow). (Best viewed in color.)

The landmark-specific neighborhood sets now present the structure elements of the graph model over which, the MRF-like process is run.

1.2.3 Landmark-Specific Graph Definition

Every landmark i is treated in SpatialModel via a graph G_i built over the subject landmark and its neighbors set $N_u(i)$ and executed by the MRF-like process.

The vertices of each landmark-specific graph are confounded with the set of landmarks $\{i \cup N_u(i)\}$. The edges of the graph, however, are restricted to link the subject landmark vertex to its neighbors' vertices and discard every inter-neighbor relationship. Note that this measure drastically alleviated the SpatialModel complexity as it is running an iterative process proportionally dependent on the degree of the treated graph (number of edges). Based on the graphs definition, fully connected sub-graphs, also called cliques, and over which the MRF-like process should be executed, are restricted to bi-vertices graphs (i, j) where $j \in N_u(i)$

Now that the landmark-specific graphs G_is are defined over the landmark-specific neighborhood $N_u(i)$, the following adopted process and its approximation are discovered for inferring the marginal probabilities of landmarks' locations given the locations of a subset of other landmarks.

1.2.4 SpatialModel Implementation

For a landmark $i \in S$, it is denoted by $x_i = (u_i, v_i) \in \mathbb{R}^2$ the random variable associated with its spatial coordinates. It is also denoted by $p(x_i)$ the unary marginal probability indicating that a landmark i is located at the site x_i. For simplicity, it is written p_i to indicate $p(x_i)$. And by $\hat{p}_i$ its approximation by SpatialModel. [26] adopted the following potential-like function where the unary marginal probability of a landmark i is inferred

given the position of all other landmarks:

$$\hat{p}_i = \frac{1}{Z} \prod_{j \in N_u(i)} p_{i|j} * p_j + b_{j \to i} \tag{2.3}$$

Where $p_{i|j}$ is the learned conditional prior of the pairs of landmarks (i, j), $b_{j \to i}$ is a bias term used to describe the background probability for the message passing from a landmark j to i, $*$ presents the convolution operation, and Z is a normalization function that will be later discarded in the model approximation presented in (2.4). Similarly to Tompson et al. [26], the equation (2.3) is adopted and run it over every clique (i, j) $/ j \in N_u(i)$ in G_i.

The SpatialModel task could be summarized as an incremental filtering process of the LandmarkDetector assumption against its predefined neighbors to consolidate or inhibit this assumption as detailed below:

First, lsp is defined as the log of the Softplus equation where, $\text{Softplus}(x, \beta) = \frac{1}{\beta} \ln(1 + e^{\beta x})$. lsp is applied to p_i to convert into an initial marginal energy $me_i = lsp(p_i + \varepsilon)$ such $\varepsilon > 0$. Then, for every predefined conditional landmark (also called neighbor) in $N_u(i)$, $p_{i|cond_i}$ is convolved with p_{cond_i} after being Softplus transformed and before being log-transformed. The acquired quantity is iteratively added to the initial quantity me_i. The final quantity is exponentially transformed to return from the log transform space initially applied. The described process can be summarized by the following equation:

$$\hat{p}_i = \exp\Big(me_i + \sum_{cond_i \in N_u(i)} \ln[\text{Softplus}(p_{i|cond_i}) * \text{Softplus}(p_{cond_i}) + bias + \varepsilon]\Big) \tag{2.4}$$

Note that (2.4) does not quantify a probability anymore due to the bench of approximations that were applied to (2.3). For simplicity, the $\hat{p}_i$ annotation is preserved to indicate the SpatialModel output. The outer multiplication in (2.3) is substituted with the log-space addition that controls the scale of the resulting quantities and hence improves the numerical stability. In addition, the Softplus function is introduced to maintain a strictly greater than zero convolution output avoiding numerical issues for input quantities of the log stage. The $2D$ convolution in (2.4) can be perceived as an incremental update of a landmark's position by its neighbors. The update's level is relative to the degree of agreement between the intensity at the estimated landmark location in p_i to their corresponding $p_{i|j}$. In other words, SpatialModel searches for the best landmark's location that agrees simultaneously with the LandmarkDetector estimation as well as with its neighbors based on their relative conditional probability $p_{i|j}$.

For a better adaptation to the LandmarkDetector model, the SpatialModel has been implemented as convolutional as possible. The provided grouped convolution function by PyTorch is leveraged to convolve simultaneously each learned conditional distribution to their corresponding estimated one.

Similarly to LandmarkDetector, it is worth stressing that the SpatialModel also produces landmarks-specific heatmaps revealing their locations. However, only the coordinate of their maximum is introduced to the learning process combined with the

LandmarkDetector's output to build a heterogeneous loss function introduced in Section 1.3. Therefore, the heatmaps positions and their spatial validity are simultaneously evaluated after being provided by the predicted LandmarkDetector's heatmaps and the Cartesian coordinates of the SpatialModel respectively.

1.3 Loss Function

As proposed earlier, the adopted loss function is a combination of two terms. The first one evaluates the LandmarkDetector's accuracy for estimating landmarks' position. A penalization term reflecting the loyalty of the detected landmarks to the learned anatomical constraints is annexed to the previous one. In other words, it is at the loss function level that one can experience the LandmarkDetector − SpatialModel combination.

For the first quantity, the adaptive wing loss function from [69] is employed to compare the predicted heatmaps by the LandmarkDetector to the ground truth. Though, commonly, the Mean Square Error (MSE) is exploited to compare two heatmaps, the urge to distinguish between the background (far away pixels from the landmark) and the foreground pixels (pixels in the vicinity of the landmark) was argued in the last reference for successful training. MSE, as a distance-based loss function, will produce low error values whenever the mass background pixels are satisfactorily located regardless of the model's performance on the foreground areas which in general, produces fuzzy heatmaps around the ground-truth locations. The robustness of the adopted function, however, is inherited from the duality in dealing with foreground and background pixels. This function is carefully designed to increase the influence of the foreground error on the overall error as long as the latter is too high. This influence will then rapidly decrease when the model is close to convergence i.e. error is within the predefined tolerances. On the other hand, the influence of background pixels is linear to the overall error that is, the focus is reduced on the background error once the foreground error is small. The AWing loss between two pixels values y and $\hat{y}$ is defined as follows:

$$\text{AWing}(y, \hat{y}) = \begin{cases} \omega \ln(1 + |y - \hat{y}|^{\alpha - y}), & \text{if } |y - \hat{y}| < \theta \\ A|y - \hat{y}| - C, & \text{otherwise} \end{cases} \tag{2.5}$$

Here, y and $\hat{y}$ are the pixel values of the ground truth heatmap and the predicted heatmap respectively. α, ω, ϵ, and θ are positive values, which were assigned to the values suggested in the original paper, i.e., 2.1, 14, 1, 0.5 respectively.

$$A = \omega \frac{1}{1 + \frac{\theta}{\varepsilon}^{(\alpha - y)}} (\alpha - y) \left(\frac{\theta}{\varepsilon}\right)^{(\alpha - y - 1)} \frac{1}{\varepsilon}$$

$$C = \theta A - \omega \ln\left(1 + \left(\frac{\theta}{\varepsilon}\right)^{\alpha - y}\right)$$

The previous parameters were carefully defined to make the loss function continuous and smooth at $|y - \hat{y}| = \theta$. Note that the exponential term $\alpha - y$ adapts the shape of the loss function to y and smooths the function a 0.

Without loss of generality, the AWing loss between two heatmaps and each containing N pixels is re-defined as the mean of all their pixel errors as follows:

$$\text{AWing}(hm_1, hm_2) = \frac{1}{N} \sum_{(y_1,y_2)\in hm_1\times hm_2} \text{AWing}(y_1, y_2) \tag{2.6}$$

Coming to the second part of the loss function runs the mean square error MSE between the SpatialModel prediction and the ground truth landmarks coordinates. Even though, as argued by Wang et al. [69], MSE is by no means the optimal loss function for heatmaps regression, it is worth stressing that the calculated error reveals the consistency metric rather than accuracy. In this case, the gradient linearity of MSE enables treating inconsistencies according to their magnitude. While this linearity is disapproved for accuracy evaluation as it leads to convergence even when many pixels still have small errors, small inconsistencies, on the other hand, should not cause the same effect on the overall performance of the model. Based on the validation error of the LandmarkDetector model vs. the mentioned combination, one could claim that enforcing facial parts constraints incites the model to enhance its prediction. The aim behind SpatialModel integration is not about seeking higher accuracy (which should be guaranteed by the CNN alone) rather than about detecting and penalizing false positives. The joint learning of the consistency and the accuracy is translated by the following heterogeneous loss function that combines the provided metric of SpatialModel with the LandmarkDetector estimations.

The final loss quantity of a predicted landmark and its ground truth is summarized as follows:

$$\text{Loss} = (1-\beta)\text{AWing}(hm, hm_{ld}) + \beta|x - x_{sm}|_2^2 \tag{2.7}$$

Where β is set to 0.1. The previous equation presents a weighted sum between the AWing's output comparing the ground truth heatmap hm to the predicted one by LandmarkDetector, hm_{ld} on one side, and the square of the L_2 distance between the ground truth landmarks coordinates x and the predicted coordinate by SpatialModel, x_{sm} on the other side. The optimization of this loss function is performed via the ADAM algorithm briefly explained in Section 1.

2 Pupil Edge Segmentation

This section presents a novel algorithm for estimating pupil size from eye-centered images. As a result of the detected facial landmarks addressed in Section 1, eye-centered ROI could be successfully cropped from the face and background input images. The proposed algorithm will be executed over the collected ROIs to extract the pupil outline which then will be fitted to the best-fitting ellipse for pupil size estimation.

The following diagram indicates the different processing steps of the proposed algorithm that will be individually described in the course of this section.

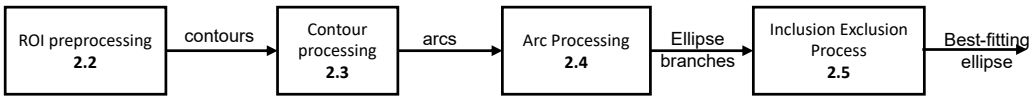


Figure 2.4: Pupil Edge Segmentation main steps

2.1 Frames' Assumptions

A set of assumptions were made about the treated eye-centered ROIs. More concretely, the ROI is assumed to present monocular images containing at most one pupil. If it exists, the pupil can be characterized as a distinctive dark blob with a radius that falls within a predefined range. It is worth noting that the characterization of the dark blob does not impose any high level of homogeneity on the pixel gray values. All common artifacts that could occur in this context such as corneal reflection, partial occlusions by the eyelashes, or any other common type of noise that could affect eye images must be permitted and correspondingly dealt with. An example from the treated images is illustrated in Figure 2.5a taken for the data set presented in [10].

2.2 ROI Preprocessing

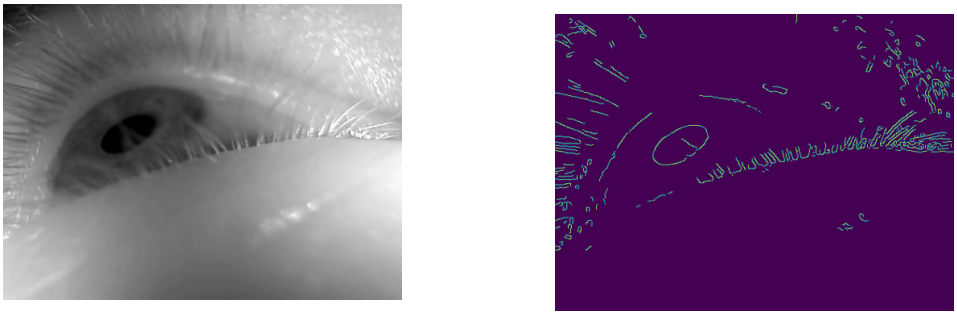
Before launching the pupil-edge segmentation algorithm, the cropped ROI undergoes a preprocessing procedure to generate the BW edge map. This procedure starts with a gray-level conversion of the input ROI. Then a 5×5 Gaussian blurring is applied to the gray-level image followed by a horizontal and vertical gradient computation which will be useful for later processing steps. Afterward, the canny edge detection and the edge thinning procedures are performed to generate the BW edge map (see example in Figure 2.5b). Finally, it has opted to extract edges as independent entities represented by their coordinates which are called contours. Every contour is represented by a list of its corresponding (x, y) coordinates presenting the outcome of this processing phase.

2.3 Contour Processing

The generated ROI from the preprocessing step is a BW thinned edge map including several edges usually originating from eyelashes, the iris, the pupil, the sclera, and other facial parts... Note that due to occlusion, inhomogeneous illumination conditions, reflection, and other irregularities; the generated shapes suffer from discontinuities. Discontinuous edges often show poor curvature conditions and even unexpected sharp turns since they are made of too-straight portions instead of continuous smooth curves. Moreover, edge curvature is systematically affected by the image resolution. Finally, the detected edges are numerous and often branched including junctions affecting their processing complexity.

This section aims to preserve pupil contours and exclude all other false positives. In addition, the presentation of the remaining edges[2] should be enhanced to prepare

[2]the word edge and contour are used interchangeably



(a) Example of the eye-centered ROI (b) Example of the treated edge maps

Figure 2.5: The input Eye ROI and the corresponding edge map. (Best viewed in color.)

their merge in the following algorithm sections. It is worth noting that due to the high number of generated contours, careful handling of their processing workflow is required by running complex operations only in the late stages of the exclusion-inclusion procedure. In the following, the different steps of edge preprocessing are briefly presented. It is aimed to perform a first round of edge exclusion and equip the remaining ones with primary features indispensable for more sophisticated features:

1) junctions and endpoints detection:

It is started by localizing and omitting the junctions in edges via image convolution with handcrafted kernels (see Chapter A). Similarly, endpoints (heads and tails) for each contour are localized via a second specialized set of kernels.

2) neighbors sorting

Neighbor sorting is a procedure that rearranges contours' points indices in a way that spatial neighbors are also index neighbors. More concretely and starting from the contour's head, the next index neighbor is the closest pixel to this starting point belonging to the contour. This procedure is incrementally applied until the contour's tail is reached. Note that by default, the pixel index within a contour is attributed based on its appearance relative to the up-left image coordinate system leading to inconsistency between indices and the spatial neighborhood. This new presentation of the contour is very efficient for the later arc processing more concretely in the arc classification process.

3) dominant points extraction

This technique replaces curves with a minimum number of representative points that preserve their original shape and geometrical characteristics drastically alleviating the amount of the treated data. As reported by Nguyen et al. [70], this technique can be addressed by two main categories namely direct and indirect methods. The first category directly relies on the edge curvature information to extract the points with locally high curvature values, whereas indirect methods fit a polygon to the subject curve that has the

least number of vertices and respects a preset level of accuracy. Dominant points were extracted from the edges by adopting the OpenCV implementation of the well-known Ramer-Douglas-Peucker Algorithm from the first category. Note that every contour generating less than 5 dominant points will be discarded due to its inadequacy to run the ellipse fitting process.

2.4 Arc Processing

The reduced contours collected from the last section acquired a set of intermediate features enabling their transformation into arcs to perform eventually their merge or exclusion. Arc transform follows the suggested procedure in [71]. However, their original algorithm is altered in two main points. 1) a contour could be transformed into more than one arc and 2) same-class combining is allowed. Arcs at this level already inherited the head and tail features from the contours explained in the previous subsection and are transformed into L and R points indicating the left and right endpoints respectively. Based on those features, arcs are equipped with more sophisticated ones such as the midpoint, the bounding box, the direction, the convexity, and the class as will be discovered in the next subsections.

2.4.1 Arc Midpoint and Bounding Box

The arc's midpoint is simply the point with the middle index of the corresponding neighbor-ordered edge. On the other hand, the bounding box BB of an arc is defined as the rectangle having the arc's L and R pixels as nonadjacent vertices.

2.4.2 Arc Direction

An arc possesses a positive or a negative direction based on its pixels' direction. The pixel's direction, in turn, is determined by the $sign(dx.dy)$ function, where dx and dy are the horizontal and vertical pixel's Sobel gradients respectively acquired from the previous section.

$$ArcDirection = \begin{cases} +, & \text{if } (dx.dy) > 0 \text{ for every pixel in Arc} \\ -, & \text{otherwise} \end{cases} \tag{2.8}$$

Every inversion of the sign will result in a split of the processed arc into two sub-arcs to continue the treatment of the non-processed part. Since the treated arcs are sorted by neighbors, the direction determination should generate continuous arcs with a homogenous direction. Note that the length criterion is applied to every newly generated arc to exclude the too-short ones (< 5).

2.4.3 Arc Convexity

Two types of convexity are defined based on comparing the upper surface U and the lower surface L delimited by the subject arc and their bounding box BB as illustrated in Figure 2.6. The convexity of an arc is determined as follows:

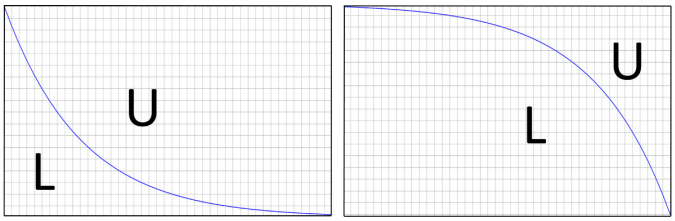


Figure 2.6: Arc Convexity determination based on its Direction & the Comparison between L vs. U Surfaces

$$ArcConvexity = \begin{cases} +, & \text{if } U > L \\ -, & \text{otherwise} \end{cases} \tag{2.9}$$

To avoid incremental computation of surface determination as in [71], it is beneficial to perceive the L surface as the integration value of the arc array along the x span. The upper one U is therefore simply the difference between the bounding box surface and the already determined surface L. The proposed algorithm in [71, Algorithm 1] could be replaced by the *trapz* function from the NumPy package.

2.4.4 Arc Classification

After having determined the direction and the convexity of each arc, a classification protocol is assigned based on these two features. Four classes 1, 2, 3, 4 are defined based on the combinations of the aforementioned direction and convexity features following the following conditions:

$$ArcClass = \begin{cases} 1, & \text{if } (+,+) \\ 2, & \text{if } (-,+) \\ 3, & \text{if } (+,-) \\ 4, & \text{if } (-,-) \end{cases} \tag{2.10}$$

Where (d,c) are the direction and the convexity signs respectively. It is worth noting that the arc classification should drastically alleviate the arc-merging process and transform it into a one-feature decision instead of exhaustingly treating all different combinations as explained in the following subsection.

2.4.5 Arc Merging

Arc pairs are merged in light of building more generalizing entities that better define the ellipse branches. This process iteratively reconstructs the pupil outline based on neighboring arcs to overcome their cracks usually due to occlusion by eyelashes or by light reflections. Note that Arc pairs will also reduce the number of the ellipse fitting process and reduce the complexity of the overall algorithm. Two arcs having two different classes (a, b) are merged if one of the following conditions is met:

$(1,2) \cap 0 \leq L^a x + tol - R^b x \leq 5tol$

$(2,3) \cap 0 \leq L^a y + tol - L^b y \leq 5tol$

$(3,4) \cap -5tol \leq R^a x - tol - L^b x \leq 0$

$(4,1) \cap -5tol \leq R^a y - tol - R^b x \leq 0$

Note that differently from the adopted version in [71], an upper bound for every condition is defined to exclude the too-far merging. In addition, the merging rules are extended with intra-class rules allowing the combination of two arcs with the same class as follows:

$(1,1) \cap 0 \leq R^b y + tol - L^a y \leq 5tol \cap 0 \leq L^a x + tol - R^b x \leq 5tol$

$(2,2) \cap 0 \leq L^a y + tol - R^b y \leq 5tol \cap 0 \leq L^a x + tol - R^b x \leq 5tol$

$(3,3) \cap 0 \leq R^a y + tol - L^b y \leq 5tol \cap 0 \leq L^b x + tol - R^a x \leq 5tol$

$(4,4) \cap 0 \leq L^b y + tol - R^a y \leq 5tol \cap 0 \leq L^b x + tol - R^a x \leq 5tol$

After having built arc pairs out of the remaining arcs, these are merged following only the arc classes' condition and without any consideration of their heads and tails position. This process is called chain search generating every continuous trajectory based on the arc pairs indices as explained in Figure 2.7. Finally, every generated chain from the last process is defined as an ellipse branch.

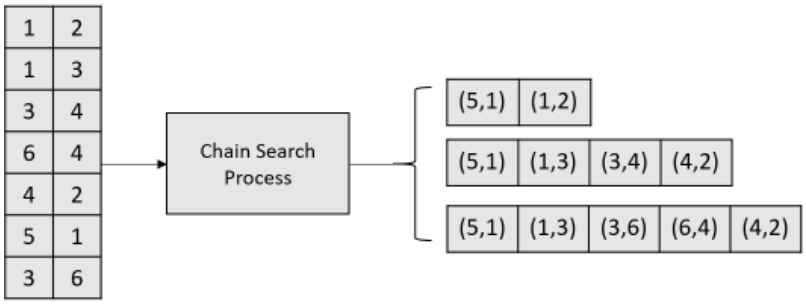


Figure 2.7: Building arc chains out of arc pairs indices: the input arc-pairs indices (left) is transformed into arc chains (right)

2.5 Ellipse Inclusion-Exclusion Process

The generated branches in Section 2.4 will be individually fitted to their best-fitting ellipse. However, one of the main assumptions as presented in Section 2.1, is that at most one pupil could exist within the treated frames. Therefore, an Ellipse-exclusion strategy must be set to exclude the resulting false positives. An ellipse is defined by its center $c = (x_c, y_c)$, its major and minor axes a and b respectively, and the angle θ. Every detected ellipse will undergo a validity check as detailed below:

2.5.1 Validity Check

An estimated ellipse is valid if the following three conditions are met:

i) the aspect ratio b/a must be greater 0.2 to exclude the too "thin" candidates as they are very unlikely would present a human pupil.

ii) the major axis a must lay within a preset range $[lower_l, upper_l]$ where $lower_l$ and $upper_l$ are a preset lower and upper limits respectively.

iii) the estimated ellipse center c must lay within the treated frame.

Even though ellipse validation will help exclude many false candidates, usually another set of ellipses will survive this check. Therefore, it is useful to compare the remaining candidates with each other to keep only one candidate according to the pupil uniqueness assumed about the treated images.

In the following, the ellipse saliency measurement technique is introduced. It is based on four different parameters to be later combined via linear regression.

2.5.2 Ellipse Saliency

In the Item d) from section 6.3.1 in Chapter 1, different saliency measurements are presented for ellipses based on various techniques. The aim behind measuring the saliency of an ellipse is to quantify its goodness relative to the preset assumptions that were assumed about it. In the work, ellipse saliency is addressed by four metrics: The distance of the ellipse outline to their fitting points, the degree of agreement between the ellipse gradient and the image gradient at the fitting points, and the angular spread as explained below:

1) ellipse-point distance

The closeness C of the fitting points to the estimated ellipse can indicate the quality of the subject estimation. The closer the points are to the estimated ellipse, the better the estimations. Therefore, an ellipse-point distance technique is addressed to quantify this closeness. Two main techniques are found in the literature that could calculate the distance whether iteratively or directly (the adopted technique).

i) iterative calculation of the ellipse-point distance

The point-to-ellipse distance equation $F(t) = (\frac{e_0 x}{t + e_0^2})^2 + (\frac{e_1 y}{t + e_1^2})^2 - 1$, where x and y are the coordinates of the point and the ellipses' width and height are $2e_0$,$2e_1$ respectively. This function is generally a fourth-degree one (except in the case of circles, $e_0 = e_1$). Three main methods are used to solve $F(t) = 0$ namely: the bisection method, the Newton method, or the polynomial method, an extensive reference for these methods and their implementation can be consulted in [72]. Even though these methods differ in their analytical representation, they all share the fact of being solved iteratively.

ii) direct calculation of the ellipse-point distance

Świrski et al. [10] adopted an explicit error function designated by Error of Fit 2*nd* order EoF_{2px} introduced by the function below:

$$EoF_{2px} = \frac{Q(x_i, y_i)}{|\nabla Q(x_i, y_i)|} \frac{|\nabla Q(x_{1px}, y_{1px})|}{Q(x_{1px}, y_{1px})} \tag{2.11}$$

Where, Q is the ellipse function and $\nabla Q = (\nabla Q_x, \nabla Q_y)$ its gradient. The closeness of a point $p_i(x_i, y_i)$ is compared to the closeness of one pixel away from the minor axis p_{1px} calculated by the EoF_{2px} function. To better illustrate the behavior of this error function Figure 2.8 shows the error value of pixels both inside and outside the ellipse presented by its black contour.

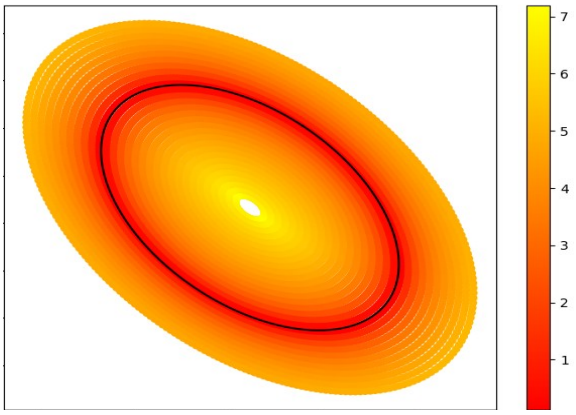


Figure 2.8: EoF_{2px} behavior for pixels both inside and outside the ellipse

Therefore, EoF_{2px} is adopted to calculate the cumulative error of the fitting points as an indicator of the overall closeness C of the ellipse assumption to them.

$$C = \sum_{p \in inliers} EoF_{2px}(p) \tag{2.12}$$

2) gradient agreement

This metric considers the agreement A between the image gradient and the ellipse gradient of a point in the inliers set.

$$A = \sum_{p \in inliers} \frac{(g_x \nabla_x Q + g_y \nabla_y Q}{2} \tag{2.13}$$

Where g and ∇Q are the image gradient and the ellipse function gradient respectively. Even though A is directly dependent on the length of the inliers subset, it is rewarded by 1) points where their image gradient agrees with their ellipse gradient 2) points with a high

image gradient value which is following the hypothesis that a pupil gradient must lay on the strong edge. It is believed, as a consequence of the arc conversion procedure, that 1) becomes less relevant as it is way less likely to happen. In other words, the arc-based fitting procedure guarantees to some extent the gradient agreement between the image circumstances and the proposed model.

3) angular spread

This metric quantifies the angular spread S of the ellipse branch relative to the ellipse hypothesis computed for every arc within the subject branch separately. More concretely, the angular spread of an arc is the angle made by the estimated ellipse center and the L and the R points of the arc a. Finally, the angular spread of the branch b is the sum of the arcs' angular spreads.

$$S = \frac{1}{360} \sum_{a \in b} \cos^{-1}\left(\frac{\overrightarrow{C_b L_a}.\overrightarrow{C_b R_a}}{|C_b L_a||C_b R_a|}\right) \tag{2.14}$$

2.6 Best-Fitting Ellipse

The decision of the best-fitting ellipse is made based on the saliency scores calculated for every "valid" ellipse. Note that an ellipse saliency is measured over three different parameters as explained in Section 2.5. Furthermore, every score belongs to a different value range making their combination, a not straightforward task. Therefore a binary-logistic-regression model training is launched after having gathered the mentioned sanity parameters for all valid ellipses and by assigning 1 to the correct ellipse and 0 to the false ones. Note that a valid ellipse is a correct one if its center is less than 5 pixels from the ground truth, otherwise, an ellipse is labeled as a false one.

$$Label = Sigmoid(\alpha C + \beta A + \gamma S) \tag{2.15}$$

Where α, β, and γ equal 0.2121, 6.4256, and -9.9787 respectively.

2.7 Depth Integration

After segmenting the pupil using the best-fitting ellipse method, the need to account for the depth variability caused by the BFP circumstances emerged. In this context, it is meant by depth the distance between the $3D$ object and the camera center. Depth variation during the measurement must affect the mm/pixel ratio that indicates the spatial quantity captured by the pixel. To achieve this, the depth information was integrated into the data allowing to correct for any differences in the estimated pupil size in the previous subsection of the analysis. The conventional approach for estimating depth relies on information obtained from stereo imaging using two cameras. In the proposed concept of BFP, this information is readily available. Therefore the proposed solution is designed

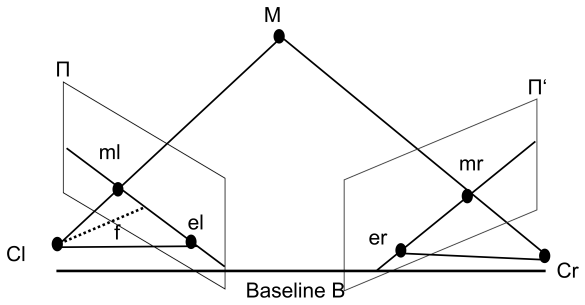


Figure 2.9: M: 3d object, m: $2d$ projection of M on the optical plan, π, π': camera plans, C_r, C_l: camera center, B: baseline distance between Camera centers, e: the epipole [3], L: epipolar line [4], f: focal length.

for these circumstances. The depth information is derived from the so-called disparity d parameter. Broadly speaking, disparity reveals the depth's information out of the spatial shift of a $3D$ object projected on each image from the stereo view system. In Figure 2.9.

In the following, an end-to-end demarch for pixel depth estimation is provided. The different steps are realized via predefined functions from the OpenCV library.

2.7.1 Camera Calibration

Camera calibration aims to determine the camera's intrinsic and extrinsic parameters permitting to link of the pixel information in the frame to the $3D$ world. Initially, these parameters are determined based on the well-known pinhole model as follows: Extrinsic parameters or camera pose are essentially figuring out camera position and orientation in a $3D$ space. In addition, It allows us to map from $3D$ world coordinates to 3D camera coordinates and vice versa. It is composed of a rotation component and a translation one which together build a matrix that can be expressed by $[R, t]$. Intrinsic parameters, however, allow us to map from the $3D$ camera coordinates to the $2D$ image coordinates. This mapping is expressed with the so-called intrinsic matrix K and presented as follows:

$$K = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.16}$$

With (u_0, v_0) are the coordinates of the principal point of the camera sensor, α and β are the scale factors along u and v axes respectively, and γ describe the skewness (a parameter that reflects orthogonality of sensor axes relative to the optical axis) of the two image axes. These two matrices link the real-world frame to the image frame as follows:

$$sm = K[Rt]M \tag{2.17}$$

[4]intersection of C_rC_l with the image planes

[4]the line connecting the epipole e with the $2D$ point m

s is an arbitrary coefficient indicating that all points issued from the same light ray will project on the same point in the image.

Note that parameter estimation follows the proposed algorithm by Zhang [73] that finalizes the closed solution by a second-order optimization for lens distortion reduction and can be achieved via the proposed function *cv.calibrateCamera* from OpenCV.

2.7.2 Fundemental Matrix Estimation

The process of estimating the Fundamental matrix starts by finding a set of matches also called stereo correspondence in both images generally achieved separately via well-established methods (windowed correlation, SIFT descriptor). In this case, stereo correspondence will be a straightforward task since facial landmarks are detected in both images and are naturally matched. The fundamental matrix F is a mapping that projects a point m from the first view (image) to the epipolar line l' of the second view following the equation below:

$$m^T F m' = 0 \tag{2.18}$$

The estimation of the fundamental matrix is a crucial step in the process of determining the epipoles in each view. Once the fundamental matrix is calculated, the next step is to find the intersection of two or more epipolar lines (depending on the practical context). This process allows us to estimate the epipoles with a high degree of accuracy. The OpenCV function used for fundamental matrix estimation is called *cv.findFundamentalMat*

2.7.3 Image Rectification

Rectification transforms epipolar lines into parallel and horizontal lines meaning m_l and m_r will lay on the same vertical level in each view. This transform is realized by finding two homographies H_r and H_l such that the epipoles of the left and right view e_r and e_l are projected to $(1, 0, 0)$ points. For a comprehensive algorithm about image rectification for the stereo image system, I suggest Monasse et al. [74] as a reference.

2.7.4 Disparity

Once both views are rectified, the disparity between m_l and m_r, which is the spatial shift between them is inversely proportional to the depth of their corresponding point M. The following formula describes quantitatively this relation.

$$Z = f.\frac{B}{\delta(x)} \tag{2.19}$$

with f the focal length from the camera calibration process, B is the base line and, $\delta(x) = x_r - x_l$ the coordinates of m_r and m_l respectively.

3 Accommodation Reflex Investigation

The accommodation reflex represents the process of lens thickening for focusing on near objects. Considering that pupil constriction also occurs during accommodation, it is wise to detect this phenomenon when specifically seeking a light-dependent pupillary light reflex (PLR) signal, which is referred to as the objective PLR.

Since this phenomenon is generally associated with eyeball convergence as explained in [75], one can adopt this feature for exploring accommodation. Once convergence is detected, the proposed algorithm performs a pupil size investigation on the last frames taking place before this convergence to confirm the investigated accommodation. This valuable information can then assist the technician in suspecting the occurrence of accommodation, enabling them to interrupt it or filter the affected data in a post-processing step which should enhance the objectivity of PLR. Note that the two-step accommodation investigation algorithm is built to take advantage of the previously achieved features, to facilitate its integration into the central processing stream of the BFP system. More concretely, the proposed algorithm takes advantage of the detected facial landmarks and the segmented pupil edge features presented in Section 1 and Section 2 respectively.

The gaze direction G is continuously estimated and compared to a threshold th_g to determine if the eyeballs are converging. Based on this analysis, the next step involves calculating the velocity of the pupil constriction CV. Once $G < th_g$, CV is then compared to a threshold value th_{cv} to decide whether a potential accommodation event is happening. In Figure 2.10, an illustration of the accommodation investigation algorithm is presented.

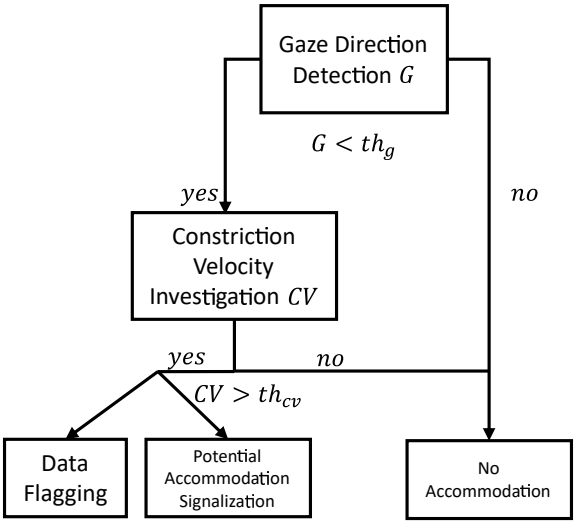


Figure 2.10: The accommodation investigation algorithm

3.1 Eyeballs Convergence Detection

It is meant by eyeball convergence in this work when the Line of Sight (LoS) of both eyes converges to keep the near object aligned with the fovea of each of them to produce a sharp and focused vision. The direction of LoS is called gaze and the process for determining this direction is called gaze estimation. In this work, the gaze direction is represented by $g = (\theta, \phi)$ where θ and ϕ designate the pitch and the yaw directions respectively. In [76], the LoS is formally defined as the straight line passing through a $3D$ point within the object, the fovea, and the optical center of the eye lens. In this work, however, the LoS is imitated by the unitary vector originating from the pupil center and having g as a direction. Therefore, eyeball convergence will be primarily addressed with gaze direction estimation as will be detailed below. In Section 6.4.3 from Chapter 1, an introduction to gaze estimation methods with a focus on the appearance-based category was provided which is also the adopted category.

Appearance-based methods are affected by head-pose variation. Hence gaze estimation should benefit from integrating this information into the input data as was argued in many studies. Head-pose integration could be executed explicitly or implicitly. In explicit integration cases, models generally incorporate eye images and append the $3D$ head-pose vector into their training process. Whereas in the implicit integration cases, models incorporate face images that inherently encompass head-pose information. It is worth noting that the explicit integration as in [57] or later in [77] showed lower performance compared to implicit integration models as in [78]. A more extensive comparison between both categories is provided in [61, Table V].

To take advantage of the available information provided from the previous processing steps, and given that inherently integrated head pose information enhances the gaze estimation performance, a carefully designed input signal is built to be later fed to the adopted model for gaze determination. In the following, this input signal also called Intermediate Representation is introduced, by depicting the transformation process applied to the previously detected facial landmarks and the extracted pupil layout. Thereafter, the adopted model also called gaze estimator, and the training dataset are presented in Section 3.1.2 and Section 3.1.3 respectively.

In Figure 2.11, the diagram revealing the different processing steps is presented for the subject gaze estimation algorithm. In the next subsections, each processing step is individually presented

3.1.1 Intermediate Representation

Adopting an intermediate representation instead of directly regressing eye images or face images was argued by the complexity of the mapping resulting from such high-dimensional input as in [60]. The authors were based on the fact that performance improves for eye-image-based algorithms by simply adopting a larger model. Adopting a simpler presentation of the input signal is hence achievable with a lower-complexity neural network and without any loss of performance. Therefore, and in light of these arguments, an intermediate representation of the eye and the head pose is created to

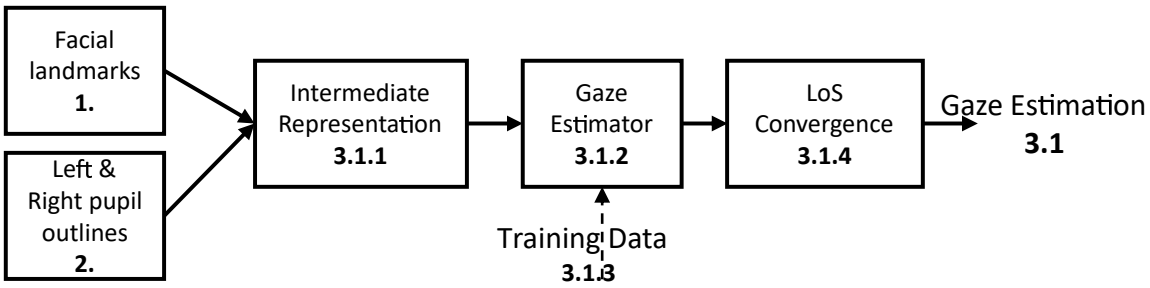


Figure 2.11: The gaze estimation diagram, numbers at each box indicate the corresponding section.

perform $3D$ gaze regression. These low-dimensional binary maps were built based on the main workflow of the BFP system as will be explained above:

i) Facial-shape binary-map

The first binary map is constructed over the detected landmarks addressed in Section 1 to build a facial layout as symbolized by the blue shape in Figure 2.12. The aim behind it is integrating the *valuable* head-pose information into the input signal.

ii) Eye binary-map

Two binary maps are built over the extracted left and right pupil outlines and presented by the best-fitting ellipse as was addressed in Section 2. They are respectively symbolized by the turquoise and magenta ellipses in Figure 2.12.

The intermediate representation is therefore formed by the superimposition of the three binary images forming a 3-channel heat map that encompasses the above-mentioned components. Note that the pupil scale was augmented to fit the expected elliptical shape at the provided spatial resolution. As scale should not influence the gaze direction, this procedure must help the pupil signal reach the latter layers of the gaze regressor by surviving the spatial resolution deterioration after several convolutional operations.

Opposite to [60] who exploited single-eye images, the intermediate representation preserves the relative position of the pupils to one another and their relative position relative to the provided facial layout. Which in turn must provide relevant information about the eye-ball convergence.

3.1.2 Gaze Estimator

After having specified the intermediate representation characteristics, this input signal is exploited to train the gaze estimation model. The same choice as in [60] was followed by adopting the DenseNet CNN originally presented in Huang et al. [79]. DenseNet is known for achieving a comparable level of accuracy with a lower number of parameters compared to other CNNs designs. Its distinguishing property is that each layer receives additional input from all preceding ones via concatenation. More concretely, the treated

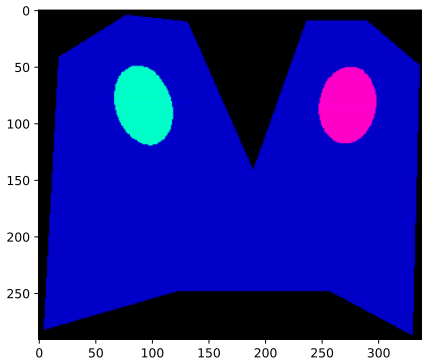


Figure 2.12: Superimposition of the left-pupil, right-pupil, and the face-shape. (Best viewed in color.)

input at each layer is the concatenation along the feature dimension of the output of all the previous ones. Due to this architecture, the *collective knowledge* gathered at each network's level could be treated with fewer parameters by avoiding to *relearn redundant feature-map.*

In DenseNets, data is processed with an alternation between DenseBlocks and TransitionLayers. In the following, a brief presentation about the characteristics of both components is introduced to disclose the data stream within DenseNets.

i) DenseBlock

The DenseBlock is a succession of DenseLayers formed by a 1×1 convolution also called bottleneck followed by a 3×3 convolution as clearly illustrated in Figure 2.13. Bottlenecks are introduced before the 3×3 convolutions to reduce the number of input feature maps to improve the computational efficiency. On the other hand, the 3×3 convolutions present the feature-learning layer where it receives the concatenation of all the preceding ones and adds k (also called growth-rate) new feature maps.

ii) TransitionLayer

TransitionLayers, however, consist of a 1×1 convolutional layer followed by a 2×2 average-pooling layer. The 1×1 convolution in the TransitionLayers is applied to define the number of feature maps to be treated by the next DensBlock usually regulated via a compression factor $\theta < 1$ to enhance the network's compactness. On the other hand, the 2×2 average pooling is responsible for the spatial downsampling of the feature maps and for their size matching enabling their concatenation before they are fed to the next DenseBlock. Note, that every convolution operation is preceded by a (BN-ReLu) operation.

iii) OutputLayer

The output of the last TransitionLayer undergoes a final ReLu and average pooling followed by a linear transformation to finally meet the output dimension. Note that gaze

direction is jointly estimated for both eyes since the input signal for the gaze regressor should integrate information simultaneously about them.

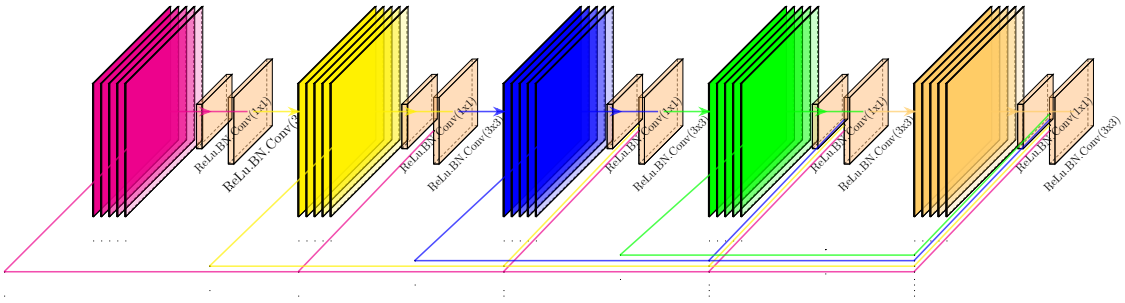


Figure 2.13: An illustration of a DenseBlock with 5 DenseLayers (Bottleneck + 3×3 convolution). (Best viewed in color.)

The same hyperparameters adopted by Huang et al. [79] are employed to regress the $3D$ gaze direction such $k = 8$, $\theta = 0.5$ and 5 DenseBlocks each containing 5 layers. It is due to the simplicity of the input space (presented in Section 3.1.1), that such a lightweight network could be afforded. In fact and opposite to eye images, the channels in the intermediate representation present only binary values giving rise to a well-defined background-foreground identification.

3.1.3 Dataset

To train the gaze estimation model, the recent gaze direction data sets were investigated, as introduced in [61, Table 4], to check their eligibility for building the intermediate representation introduced in Section 3.1.1. It was concluded that the End-to-end Video-based Eye-tracking data set also called (EVE) and introduced by Park et al. [80] meets the specifications in terms of the provided data annotations required for building it.

The EVE dataset provides 68 different facial annotations as well as the left and right pupil sizes in *mm*. In addition, the mm-to-pixel ratio is also provided which enables a direct conversion of the pupil size in pixels. Based on the provided annotations, the proposed intermediate representation could be built by finding the best-fitting ellipse for each pupil and by building the facial shape encompassing the head pose information.

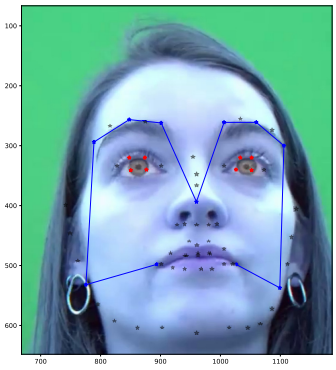
i) Facial Shape

To preserve the consistency of the gaze estimation algorithm with the main data processing stream, the facial shape in the intermediate representation was built over the same landmarks provided by the facial landmark detector presented in section Section 1. Since only 17 landmarks are estimated as presented in Figure 2.3, the equivalent annotations from the EVE dataset were selected in an attempt to mimic a similar facial shape that would be generated by the landmark detector. Those equivalent annotations correspond to the landmarks: $l_3, l_{13}, l_{17}, l_{19}, l_{21}, l_{22}, l_{24}, l_{26}, l_{30}, l_{36}, l_{39}, l_{42}, l_{45}, l_{48}$, and, l_{54} building the facial

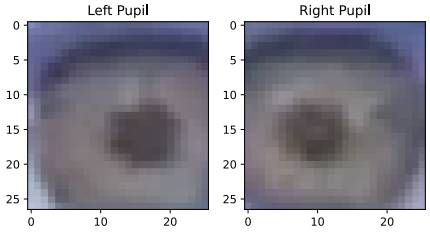
shape. The region outside the blue contour presented in Figure 2.14a is then set to a 0-valued background.

ii) Best-fitting Ellipse

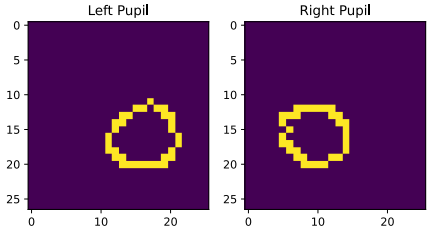
In Algorithm 1, the subsequent steps for cropping the pupil ROI based on specific landmarks were provided as illustrated in Figure 2.14b. Thereafter, the pupil edge is extracted for both eyes by following a similar procedure as in Section 2.3 and by exploiting the provided pupil size as presented in Figure 2.14c. Finally, the best-fitting ellipse for the remaining edges is found via an optimization process. It is worth noting that the initial estimation for this process is determined by Hough-based voting.



(a) Example from EVE data set, red landmarks are used for pupil extraction and able connected labels to build the facial shape. (Best viewed in color.)



(b) The segmented left and right pupils. (Best viewed in color.)



(c) The corresponding edges from the segmented pupil. (Best viewed in color.)

Figure 2.14: Facial shape and pupil outline extraction from EVE Dataset

Algorithm 1 Pupil's Best-fitting Ellipse from EVE Dataset

Data: x, l, r; image, facial labels, pupil radius
Result: y; best-fitting ellipse
$x \leftarrow gray(x)$; gray level conversion
$w_l \leftarrow dist(l_{37}, l_{38})$ $w_r \leftarrow dist(l_{43}, l_{44})$; left and right pupil widths
$l_l \leftarrow dist(l_{40} - l_{41})$ $l_r \leftarrow dist(l_{46} - l_{47})$; left and right pupil lengths
$w \leftarrow \max(w_l, w_r)$ $l \leftarrow \max(l_l, l_r)$; adopted width and length
$ROI_l \leftarrow roi(l_37, l, w)$ $ROI_r \leftarrow roi(l_43, l, w)$; left and right pupils ROIs
$ROI \leftarrow gaussian(ROI)$; 5×5 kernel
$G_h, G_v \leftarrow gradient(ROI)$; horizontal and vertical gradient
$ROI \leftarrow canny(ROI)$; canny edge detection
$ROI \leftarrow edge\_thinning(ROI)$; edge thinning
$ROI \leftarrow junction\_omission(ROI)$; junction omission
$contours \leftarrow contour(ROI)$; contour extraction
for *c in contours* **do**
  if $5 \leq len(c) \leq 10r$ **then**
    $c_{poly} = poly(c)$; polygonal approxiamtion of contour c
    if $len(c_{poly})) \geq 5$ **then**
      $h_c, t_c \leftarrow find\_extremes(c)$; head and tail of contour c
      if $dist(h_c, t_c) \leq 10r$ **then**
        $c.validity = True$
      else
        $ROI[c] = 0$; eliminate invalid contours from ROI
      end
    end
  end
end
$first\_guess = CHT(ROI, r, G_h, G_v)$; Circular HT for first-guess estimation of the ellipse parameters
$y = fit\_ellipse(first\_guess)$

3.1.4 LoS Convergence

After determining the gaze direction for both eyes, the collected estimations were transformed to rebuild the LoS as defined in Section 3.1 to check their convergence. The degree of convergence of both LoS is measured by the angle between them called α to support or refute the accommodation's hypothesis. Once α is smaller than a threshold angle α_{th} ($\alpha < \alpha_{th}$), the accommodation hypothesis will be further investigated as explained in Section 3.2, otherwise the investigation process will be discarded for the subject frame.

3.2 Pupil-Constriction Velocity Investigation

In the case where the left and the right LoS show a high convergence level, the progression of the pupil size at the previous frames is then investigated. This parameter represents the velocity of constriction of the pupil that should increase in case of eventual accommodation. If the constriction velocity CV between two successive frames exceeds a threshold value th_{cv}, the potential accommodation will be signalized to the technician, and the subject frame will be labeled as potentially accommodating in an attempt to enhance the objectivity of the collected measurements.

Note that this examination could not take place on the fast-constriction phase, since the pupil reflex is dominated by the light stimulus, and chances of accommodation are rather low.

–3–
Materials

This chapter briefly presents the different tools that were used to implement each component that was introduced in Chapter 2. It is worth noting that Python was the main programming language for this work.

On the other hand, different metrics that measure the performance of each of component as illustrated in Chapter 4, are introduced.

1 pupil detection

1.1 Packages

The present CNN in the previous chapter was implemented in PyTorch introduced by Paszke et al. [81] which is defined as an *Imperative Style, High-Performance Deep Learning Library*. Note that Pytorch provides the Conv2d, BatchNorm2d, and MaxPool2d methods which are the fundamental building blocks of the CNN model. During the model learning phase, the communication to the training data sets was done via Pandas initially introduced by McKinney et al. [82]. In this work, Pandas was used for different data augmentation, data transformation, and label management protocols.

1.2 Dataset

The adopted datasets for the CNN-based pupil detector are as follows.

1. the 300w dataset presented by Sagonas et al. [83] and was built for the first automatic facial landmark detection in-the-wild challenge (300-W 2013). The annotation are provided with 68 landamrks

2. the HELEN presented by Le et al. [84] which consists of 2000 training and 330 test images with highly accurate, detailed, and consistent 194 annotations of the primary facial components.

3. and the WFLW for (The Wider Facial Landmarks in the Wild) created by Wu et al. [85] and contains 10000 faces (7500 for training and 2500 for testing) with 98 annotated landmarks. This database also features rich attribute annotations in terms of occlusion, head pose, make-up, illumination, blur and expressions.

1.3 Data Enhancement

The previously mentioned datasets provide a significant number of annotations (e.g., 98 landmarks for WFLW, and 68 for 300w and LFPW). This could be useful for tasks such as face recognition or facial expression analysis. For facial parts localization (this study case), however, obviously, way fewer annotations are required. Therefore, the proposed model was trained with a limited yet sufficient amount of landmarks, annotating well-defined points, and preserving the global facial shape. In other words, the number of landmarks should be defined by a lower limit below which, the global facial outline will be drastically affected. The significance of this lower bound emerges from the "Sufficient Landmark Density" concept where studies analyze the dependency between the optimization quality and the number of landmarks constituting an Active Shape mode. Seshadri et al. [86] and Milborrow et al. [87] showed that ASM must be modeled with a sufficient landmark density to reach some fitting accuracy level.

The initial thoughts were to simply select a subset from the provided landmarks fulfilling the predefined needs. However, after investigating the exploited datasets, it was found that landmarks are rather forming contours around facial parts than annotating well-defined points. In Figure 3.1[1], one can notice how the labeling process is meant to annotate the upper and lower lips contours which do not necessarily result in clear mouth extremities landmarks. In addition, Nose tips as well as the pupils' centers were absent among annotations.

Consequently, the annotations of the training datasets were enhanced with an additional five well-defined landmarks: the two pupil centers, the nose tip, and the two mouth extremities. Figure 3.2 presents the dedicatedly designed user interface for the labeling process. This process generated 17 fully defined facial landmarks as illustrated in Figure 2.3.

1.4 Metrics

Different metrics were adopted for evaluating the proposed pupil detector:

1. the Percentage of Correct Key-points (PCK) Yang et al. [88]. This metric assumes that a landmark's location is correctly estimate if the distance between the predicted location and the true location is within a certain threshold.

2. The Normalized Mean Error (NME) is calculated using the L2 distance, normalized by the interpupillary distance derived from the locations of both pupils' centers.

[1]Image from the HELEN dataset, `http://www.ifp.illinois.edu/~vuongle2/helen`

Figure 3.1: Examples of inconsistent labeling where the same color should present the same facial landmark. However, the mouth extremity is presented in pink (left image) and in yellow (right image). A zoom of the region is presented in the second row. (Best viewed in color.)

3. the Cumulative Point-to-Point Error Distribution. Intuitively speaking, this metric indicates the percentage of estimations that fall below a specific fraction of the interpupillary distance.

2 Pupil-Size Estimation

2.1 Packages

As a pure image processing challenge, the proposed algorithm for pupil segmentation and pupil size estimation was mainly implemented via OpenCV which was introduced by Bradski [89].

2.2 Datasets

The proposed algorithm was achieved via the following datasets:

1. the Swirski dataset introduced by Świrski et al. [10] which contain 940 for each eye of two participants. The pupil labeling is described by the beste-fitting ellipse via its centre, major radius, minor radius and angle of the major axis with the x-axis (in radians).

2. the LPW dataset standing for Labelled Pupils in the WildLabelled Pupils in the Wild and introduced by Tonsen et al. [90]. This dataset contains 66high resolution and high speed videos from 22 participants with different ethnicities.

3. the Robust Pupil Detection in Real-World Scenarios: ExCuSe dataset introduced by Fuhl et al. [91]. The dataset consists of overall 38401 images, where the pupil position was labeled manually on each image.

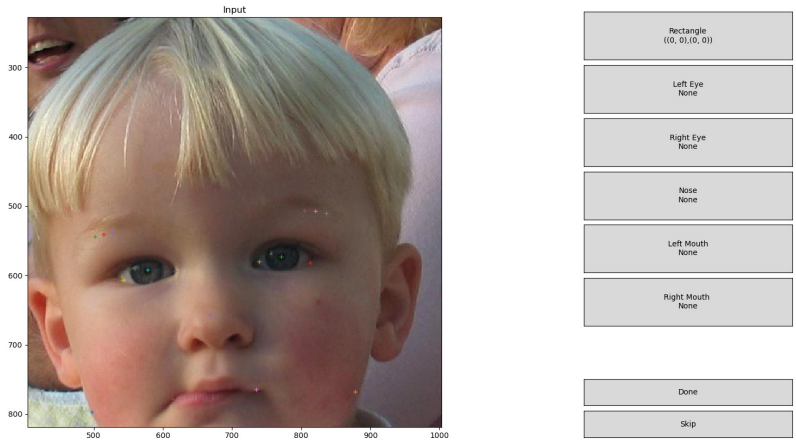


Figure 3.2: Data Enhancement Interface for Landmarks Annotation (Best viewed in color.)[2]

2.3 Metrics

Different metrics were adopted for evaluating the proposed pupil-size estimator:

1. the Cumulative Detection Rate: presenting the precentage of ellipse-center estimations that are within a specific number of pixels form the groung truth

2. the Detection Rate per folder/video at five pixels for every dataset. Based on the convention where a successful estimation is the one that is located maximum 5 pixels away from the ground truth, a detection rate can therefore be defined to evaluate the algorithm's performance.

3. the accuracy and the precision metrics based on the True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) metrics

3 Objectivity Enhancement

3.1 Packages

PyTorch was used to implement a neural network called DENSNET, which was adopted to estimate gaze direction. This estimation of gaze direction was used as the primary parameter to investigate accommodation and reflect the objectivity of the measured pupil size.

3.2 Dataset

The mentioned model was trained by the EVE (End-to-end Video-based Eye-tracking) dataset introduced by Park et al. [80]. It is collected from 54 participants and consists of 4

[2]image from the LPW dataset

camera views, over 12 million frames and 1327 unique visual stimuli (images, video, text), adding up to approximately 105 hours of video data in total.

3.3 Metrics

The proposed model for gaze direction estimation was evaluated via the angular distance metric expressed in degrees. This metric measures the anglular error between the two orientation of the estimated gaze and the groundtruth one.

–4–
Results

This chapter presents the performance of every component introduced in Chapter 2. Models and algorithm evaluations are addressed via predefined metrics to measure their ability to solve their assigned tasks objectively.

1 Pupil Detection

The performance of the pupil detection model described in Section 1 in Chapter 2 is first investigated qualitatively in Section 1.1. Thereafter, the quantitative evaluation of the model's performance is proposed in Section 1.2.

1.1 Effect of Filtering on LandmarkDetector

To better illustrate the impact of the spatial model mechanism on the initial heatmap estimations, a strong false positive signal was introduced by adding a stain located some pixels away and having the same intensity as the absolute maximum (the estimated landmark's location) of the heatmap as presented in Figure 4.1 from [92]: One could notice that the spatial model SpatialModel not only suppresses the introduced outlier but also generates a less blurry and less dilated blob around the landmark's location. Such filtered signals should enhance the training performance once introduced to the backpropagation algorithm for the LandmarkDetector training as proposed in [92].

1.2 Quantitative Evaluation

For accuracy evaluation and based on the Percentage of Correct Key-points (PCK) defined by [88], and inspired by its adaptation such in [93], it is denoted by PCKp the PCK relative to the distance between two pupils centers: Left Pupil center (LPc) and Right Pupil center (RPc) landmarks. 10% of the current interpupillary distance is defined as the threshold for the estimation success of a landmark. The PCK for landmark i at frame t is calculated as

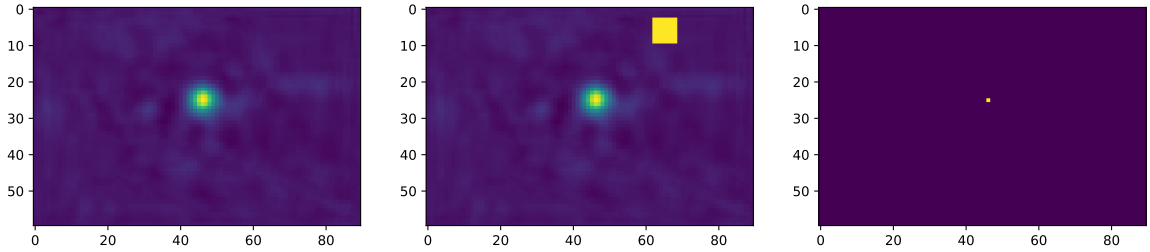


Figure 4.1: a) LandmarkDetector's Output, b) Outlier Introduction, c) SpatialModel's Output (Best viewed in color.)

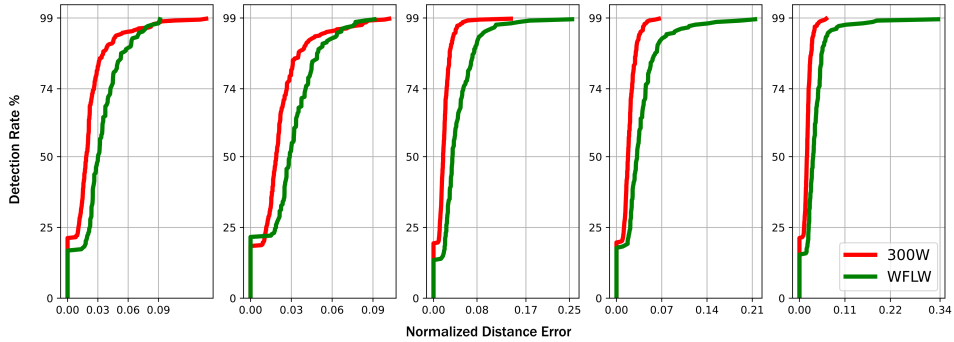


Figure 4.2: The cumulative point-to-point error distribution normalized by the interpupillary distance for LP center RP center, NT, LM extremity, and RM extremity respectively (Best viewed in color.)

follows:

$$\mathrm{PCKp}_i(r) = \frac{100}{N}\sum_{t=1}^{N} \|x_i^{t*} - x_i^t\|^2 \leq 0.1 r_t^2 \tag{4.1}$$

With $r_t = \|x_{lpc}^t - x_{rpc}^t\|$ and x_i^{t*}, x_i^t are the estimated and the ground truth landmarks locations respectively and N the number of the adopted frames.

In Table 4.1 from [92], the PCKp metric was presented for five different landmarks namely: Left Pupil center LPc, Right Pupil center RPc, Nose Tip NT, Left Mouth extremity LMe, and Right Mouth extremity RMe, for the above-mentioned data-sets.

	LPc	RPc	NT	LMe	RMe
300w	98.1	99.03	94.3	97.1	97.6
HELEN	99.0	99.4	98.4	96.4	98.7
WFLW	95.31	96.5	93.3	92.8	91.6

Table 4.1: Percentage of Correct Key-points PCKp for the main 5 landmarks for different datasets

Similarly to PCK, the median is calculated for the Normalized Mean Error (NME) which

is normalized by the inter-pupillary distance, performed for 300w and WFLW datasets and presented on the second row in Table 4.2. Even though the model's performance outperforms some approaches, it is slightly under the state-of-the-art accuracy reported by Li et al. [94] by 2.96%.

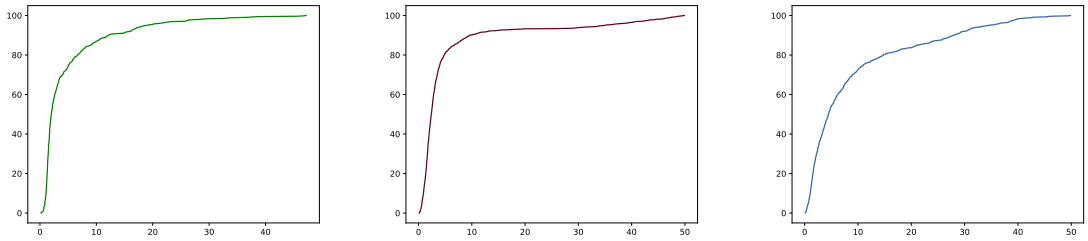
Method	NME(300w, WFLW)	NME<90%(300w)
LAB [85]	5.8, 5.27	6.5
MERGET [20]	5.29(IBUG)	4.5, 7(IBUG)
DVLN [95]	4.45, -	5.5
MTAAE [96]	4.3, 5.18	-
ODN [97]	3.56, -	9
HG-HSLE [98]	3.28, -	4.7
PIPNET [99]	3.19, 4.31	-
CE-CLM [100]	3.15, -	4.5
DTLD [94]	2.96, 4.05	-
OURS.	3.3, 4.1	4.0

Table 4.2: Median of NME for (300w+WFLW) datasets and NME for the first 90% of (300w)

To allow a more exhaustive investigation of the model's performance, Figure 4.2 from [92] presents the cumulative point-to-point error distribution normalized by the interpupillary distance for the above-mentioned five landmarks for the 300w and WFLW datasets. By comparing Figure 4.2 to the reported cumulative distributions by the references mentioned in Table 4.2, the effect of the SpatialModel in dealing with false positives could be again identified. In fact, for the 300w dataset, one can spot that on average 90% of the detected landmarks fall within 4% error. By extracting this same parameter, whenever available, from every reference in Table 4.2 from [92], one can report that all of them were overcome as summarized in the 3rd column from Table 4.2, and achieved state-of-the-art performance for this parameter. Accordingly, it could be again affirmed that the post-processing procedure helped to decrease the false positive occurrence.

2 Pupil Edge Segmentation

The pupil edge segmentation algorithm described in Section 2 in Chapter 2 is evaluated against the Swirski, the LPW, and the ExCuSe dataset presented in [10], [90], [91] respectively. The previous datasets are labeled by different parameters describing the best-fitting ellipse of the corresponding pupil. However, only the center labels were adopted for the evaluation since it is the only parameter that is present in all the mentioned data sets. In addition, these datasets are organized into separate image folders or videos where evaluations were performed on each of them separately. In the following, the set of experiments that were executed over these datasets is launched to generate different evaluation metrics.



(a) Swirski dataset (b) ExCuSe dataset (c) LPW dataset

Figure 4.3: The Cumulative Detection Rate for the different datasets (Best viewed in color.)

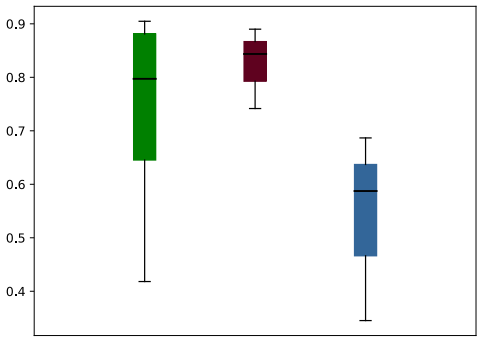


Figure 4.4: Detection rate distribution within each data set Swirski (green), ExCuSe (brown) and LPW (blue) (Best viewed in color.)

2.1 Pupil Detection Rate

A pupil is considered successfully detected if the estimated pupil center is at most n pixels away from the ground-truth pupil center. $n = 5$ is adopted to justify the human inaccuracy in the labeling process as was discussed in [12]. For every mentioned dataset, the cumulative detection rate is calculated jointly for the provided folders/videos to collect one box per dataset as presented in Figure 4.3 in [92].

In addition, the detection rate is calculated per folder/video at five pixels for every dataset described by the boxplot presentation in Figure 4.4 in [92].

2.2 Accuracy and Precision Evaluation

The second aspect that was evaluated in the proposed algorithm is the rate of false pupil estimates executed on the Swirski data set [10]. Even though the algorithm's main task is the localization of the pupil, this task is reformulated into a semi-binary classification enabling the definition of True and False estimations about positive and negative classes allowing in turn the estimations of precision and accuracy. More concretely, it is meant

by semi-binary estimations the fact that True estimations should be split into Correct and Incorrect True estimations based on the previously presented n pixels rule and as detailed as follows. In this case, the following classic fours parameters were defined as follows:

i) True Positive (TP):

represents cases in which the algorithm detects an existing pupil. similar to Santini et al. [12], the Correct True Positive (CTP) and Incorrect True Positive (ITP) classes are defined based on the 5px rule adopted in Section 2.1.

ii) False Positive (FP):

represents cases in which the algorithm wrongly detects a non-existing pupil.

iii) True Negative (TN):

represents cases in which the algorithm and ground truth agree on pupil absence.

iv) False Negative (FN):

represents cases in which the algorithm fails to detect an existing pupil.

Based on the previous parameters the accuracy and precision of the algorithm will be inspected as defined in the equation Equation (4.2) and Equation (4.3) respectively.

$$Accuracy = \frac{CTP + TN}{TP + TN + FP + FN} \tag{4.2}$$

$$Precision = \frac{CTP}{TP + FP} \tag{4.3}$$

Note that the TP parameter in the numerator is substituted with the CTP one to avoid rewarding the evaluation with ITP estimations. As one can notice negative samples (without pupils) must be provided to execute the aforementioned tests. Therefore, the Swirski data was manually set into 2 new classes 1: existing and 0: non-existing pupil. The labeling process resulted in 181 negative samples. After running the provided parameters on the Swirski dataset it could be concluded that the accuracy and precision are estimated by 76.04%, and 81.75% respectively.

3 Accommodation Reflex Investigation

The evaluation of the model for gaze estimation for accommodation investigation introduced in Section 3 in Chapter 2, is mainly performed on the EVE data set provided by Park et al. [80]. This dataset is presented by various labels describing different aspects of the $3D$ gaze direction of the eye. A lightweight Densnet is exploited to regress the $3D$ gaze direction out of an intermediate presentation clearly explained in Section 3 in Chapter 2. The performance of predicted gaze directions is assessed using an error metric

that measures the angular distance expressed in degrees to the ground truth as explained in the following equation.

$$L_{gaze} = \frac{1}{NT} \sum^{N} \sum^{T} \frac{180}{\pi} arccos(\frac{g.\hat{g}}{\|g.\hat{g}\|}) \tag{4.4}$$

The model achieved a 7.6° mean error demonstrating sufficient accuracy for detecting eye convergence. The model performance cannot be directly compared to other gaze direction estimators as The intermediate presentation relies heavily on the performance of the pupil detection model described in Section 1 in Chapter 2. therefore the model is considered a face-based model. which shows that regressing gaze direction from the gaze map should not require a complex mapping to model.

–5–
Discussions

This thesis addressed adapting pupillary light reflex to non-cooperative patients via three main components: pupil detection, pupil size estimation, and the objectivety enhancement. Equipping a pupillometry process with such features should enable more flexibility for non-cooperative patients by tolerating moderate head poses, depth variations, occlusion, and out-of-frame occurrences. These are sequentially applied to the row images to collect accurate and objectively measured pupil sizes. The inherent features that equip every component contribute to the flexibility of the BFP system towards the treated images, unlike classique pupillometers that expect pupil-centered straightforward frames to process. Note that the previous components were implemented and then tested on publically available datasets. More specifically, The proposed solution for the pupil detection task achieved a Normalized Mean Error (NME) of around 4% on the 300w and the WFLW datasets vs. the reported state-of-the-art performance achieved by [94] 2.96% NME on the WFLW data set.
As for the pupil size estimator, the proposed algorithm was tested against three different datasets namely Swirski, Excuse, and LPW, and performed a cumulative detection rate higher than 75% for less than 5 pixels error vs. 80% achieved by [12] on the Swirski dataset. Finally, the proposed solution for the objectivity enhancement component achieved a mean error of 7.6° on the EVE dataset vs. 4.41% on the same dataset using the EyeNet network by [60].

While the individual performance of each element may not surpass the classical state-of-the-art results, we disclose in the following the performance analysis concerning the assigned task.

The pupil detector, as a preprocessing step for the pupil size estimation target, was designed to enhance performance in worst-case scenarios rather than achieving a better average accuracy on the test dataset. This can be tracked in table 4.2 in chapter 4 for $NME < 90\%$ metric where the proposed solution overcomes the reported works at this metric. Additionally, in the BFP context, the estimated ROI by the pupil detector presents a preprocessing step for the pupil-size estimator. This last component applies a more in-focus image processing to the detected ROI. Therefore, even if the pupil detector would not perform a very accurate estimation, this would not drastically impact the pupil-size

estimator.

The objective of the pupil size estimator is to accurately segment the pupil edge from frames that resemble the anticipated measurement conditions during pupillometry sessions. Specifically, the algorithm is designed to work with laboratory-taken frames captured using IR sensors. However, a challenge arises when using the provided test datasets, as they consist of outdoor frames that result in strong reflections on the eye region. This situation contradicts the central hypothesis of the algorithm, which relies on the pupil contours exhibiting a strong edge in terms of gradient values. In summary, the pupil size estimator aims to detect pupil edges under specific measurements. Without loss of generality, these conditions should present any barrier to the normal circumstances of pupillometry sessions. Nevertheless, adopting such hypotheses should gradually boost the pupil-size estimator's performance once applied to the expected frame circumstances.

Finally, it is worth noting that the proposed gaze estimator achieved a lower level of accuracy compared to the method reported in [60]. This outcome can be attributed to two main factors. Firstly, the adopted Densnet in our approach has fewer trainable parameters compared to the EyeNet variety utilized in the reference mentioned earlier. Consequently, our model may be less suitable for generalization, especially when trained on extensive datasets such as the EVE dataset. Secondly, the intermediate representation introduced in Chapter 2 is designed to detect the convergence of gaze rather than providing an accurate quantification of the gaze angle. This design choice reflects a more relaxed condition, which aligns with the specific target of the eye accommodation detection.

For the price of the under-state-of-the-art performance, it was ensured that the three components are interdependent and closely interconnected. The processing steps were carefully designed with the explicit intention of mitigating the computational complexity of the entire algorithm. This objective was accomplished by modifying subsequent processes to receive inputs from previously computed features, rather than reprocessing the raw data. By doing so, the algorithm leveraged the existing information, leading to more efficient computations and streamlined data handling. This principle was clear for the pupil size estimation step when it was executed on a narrower ROI cropped around the already detected landmarks from the eye region exhibited by the pupil detection step. This measure resulted in the reduced handling of edges per frame during their inherent classification as either belonging or not to the pupil. A second manifestation of interconnectivity between the system components can be seen in the intermediate presentation process within the accommodation detection step. This presentation was based on the already detected facial landmarks and the detected pupil from the pupil detection and the pupil size estimation steps respectively. This inherent characteristic makes our proposed solution highly compatible for integration with existing pupillometry algorithms.

Notably, the time of running of the proposed solutions was not extensively investigated, as the implementations were carried out using a general-purpose programming language. Consequently, should integration into hardware be desired, a dedicated programming language specifically designed for hardware programming would be required for a fresh implementation.

As a future direction, it is beneficial to fine-tune the pupil detector on a more specific dataset leveraging the faces of infants and babies as potential users of this algorithm. However, as far as we know no open-source high-quality labeled dataset for facial landmarks exists. Therefore, building such a dataset could be beneficial and can be done gradually from the measuring sessions.

–6–
Conclusions

This work focuses on addressing the pupillometry workflow at various levels, specifically to make it accessible for very young patients who often struggle to understand the technician's instructions.

More specifically, the detection of pupils can now be performed even under free-head-pose conditions. This advancement is made possible by utilizing a landmarks-based CNN, enabling the task to be achieved using both face and background images, rather than being limited to eye-centered images. As a postprocessing step, the spatial consistency of the CNN estimations is evaluated by comparing them to the distribution of landmarks from the training data. This utilization of landmarks distribution aids in ensuring the spatial consistency of the estimations. The introduced solution exhibits robustness against variations in illumination, direction, and head pose. This robustness allows the solution to effectively adapt to the unique specifications of young patients, accounting for their spontaneous gestures and behaviors.

A novel method was developed and evaluated for accurate pupil segmentation, specifically for estimating pupil size from the detected pupil region. This method incorporates depth information as a correction factor to improve the size estimations. The proposed approach transforms the existing contours within the ROI into arcs, which can then be merged or excluded during the pupil segmentation process. By applying a customized sanity check, the method selects the best-fitting ellipse with the highest sanity score to represent the desired pupil.

The detected facial landmarks were used once again to enhance the estimated pupil size objectively. For gaze estimation, a neural network called DENSNET was utilized, which takes an intermediate representation of the input image constructed using the detected facial landmarks. This information, along with the history of pupil size changes, is combined to enhance objectivity.

It is important to note that the data processing workflow for traditional pupillometers was maintained to facilitate the potential integration of the proposed solution into existing frameworks. Lastly, the training and inference of various models are performed using standard hardware at nearly real-time fps, enabling their integration into commercially available devices.

Zusammenfassung

Die Untersuchung des Pupillenlichtreflexes (PLR) auf ein vordefiniertes Lichtstimulationsprotokoll als Indikator für den Zustand der Netzhaut und die Funktion ihrer Photorezeptoren ist nach wie vor eine gängige Praxis unter Augenärzten. Der PLR ist nicht nur ein Diagnoseinstrument, sondern kann auch therapeutische Fortschritte aufzeigen und deren Einfluss auf die vererbte Netzhautdegeneration (IRD) überwachen. Die klassische Form dieser Untersuchung setzt jedoch ein Mindestmaß an Kooperation seitens des Patienten voraus. Da sich einige genetische Mutationen schon früh manifestieren können, ist es für eine erfolgreiche PLR-basierte Untersuchung sinnvoll, diese Untersuchung für Patienten ab dem Säuglingsalter anzupassen.

Bei klassischen Pupillometern ist die automatische Extraktion der Pupillengröße mit klassischen Bildverarbeitungsmethoden möglich, da die aufgenommenen augenzentrierten Bilder eine einfache Umgebung darstellen. Die Erweiterung der Pupillographiesysteme auf ein breiteres Patientenspektrum legt nahe, die Verarbeitungstechniken mit geeigneten Werkzeugen auszustatten, um die Situation der Nicht-Kooperation zu tolerieren. Der Grad der Kooperation wird daran gemessen, ob der Patient in der Lage ist, während der Messung einen gewissen Grad an Kopf- und Augenstille aufrechtzuerhalten. Daher werden nicht-kooperative Patienten, wie z. B. Säuglinge, durch den neuen Bildraum erkannt, der mehrere Quellen der Variabilität aufgrund ihrer freien Kopfhaltung und ihres Augenverhaltens integriert.

In den letzten zehn Jahren hat sich Deep Learning (DL) im Gegensatz zu klassischen Methoden als leistungsstarker Ansatz für Computer-Vision-Aufgaben erwiesen. Dies liegt daran, dass DL-Modelle die notwendigen Merkmale aus komplexen Hintergrundbildern extrahieren können, was zu bedeutenden Erfolgen im Bereich des Computersehens geführt hat. In dieser Arbeit wird das Problem der Erweiterung der Pupillometrie zur Integration sehr junger Patienten durch die Nutzung von DL-Techniken angegangen. Dies wird durch die Bereitstellung einer End-to-End-Lösung erreicht, die die Erfassung von PLR-Informationen von nicht-kooperativen Patienten automatisiert. Erstens wird ein neuronales Faltungsnetzwerk (Convolution Neural Network, CNN) eingesetzt, um die Komplexität des Bildraums auf das traditionelle augenzentrierte Bild zu reduzieren. Zweitens schlagen wir einen neuartigen Nachbearbeitungsalgorithmus vor, der Tiefeninformationen nutzt, um die Pupillengröße auf Subpixelebene zu definieren, um genaue PLR-Messungen zu erzielen. Drittens wird ein Tool zur Entscheidungshilfe vorgeschlagen, um die Objektivität der Messungen zu verbessern. Dieses Tool liefert wertvolle Blickin-

formationen zur Einschätzung des Akkommodationsreflexes, der ein wichtiger Faktor bei der Änderung der PLR-Objektivität gegenüber dem vordefinierten Lichtreiz ist. Dies wird durch ein zweites Modell erreicht, das DL-Techniken nutzt.

Es ist erwähnenswert, dass jedes der genannten Verfahren an öffentlich zugänglichen Datensätzen getestet wurde und eine zufriedenstellende, für den Anwendungsfall angemessene Leistung aufwies. Das Modell zur Extraktion der Pupillenregion erreichte einen normalisierten mittleren Fehler (NME) von etwa 4% bei den 300w- und WFLW-Datensätzen. Der Schätzer für die Pupillengröße wurde mit drei verschiedenen Datensätzen getestet und erreichte im Swirski-Datensatz eine Genauigkeit von 76,04% und eine Präzision von 81,75%.
In Bezug auf die Lösung zur Verbesserung der Objektivität erreichte das verwendete DENSNET einen mittleren Fehler von 7,6° im EVE-Datensatz.

Durch die Integration der oben genannten Komponenten in einen Pupillometrie-Rahmen konnte eine größere Flexibilität bei der Anpassung an das Verhalten des Patienten erreicht werden. Die Struktur dieser Arbeit wird sich um drei Schlüsselkomponenten drehen: Pupillenerkennung, Pupillengrößenschätzung und Objektivitätsverbesserung. Diese Komponenten werden abwechselnd vorgestellt, wobei ihre Grundlagen in der Literatur in Chapter 1, die verwendeten Methoden in Chapter 2 und die Leistungsmessungen in Chapter 4 behandelt werden. Darüber hinaus enthält Kapitel Chapter 3 detaillierte Informationen über die bei der Umsetzung und Bewertung verwendeten Materialien. Um den Lesern umfassendere Informationen zu bieten, sind in Chapter A Anhänge mit zusätzlichem Material zu verschiedenen im Hauptteil des Textes behandelten Aspekten enthalten.

Abstract

Investigating the pupillary light reflex (PLR) toward a predefined light stimulation protocol as an indicator of the retinal state and the function of their photoreceptors remains a common practice among ophthalmologists. Apart from being a diagnostic tool, PLR could also reveal therapeutic progression and monitor its influence on inherited retinal degeneration (IRDs). However, the classical form of this examination presumes a minimum level of cooperation from the patient. Therefore, because some genetic mutations could manifest early, a successful PLR-based investigation suggests adapting this examination for patients from infancy.

In classical pupilometers, automatic extraction of pupil size is achievable via classical image processing methods due to the straightforward environment of the acquired eye-centered images. Extending pupillography systems to integrate a broader spectrum of patients suggests equipping its processing technics with convenient tools to tolerate the non-cooperation situation. The degree of cooperation is measured by the ability of the patient to maintain some degree of head and eye stillness during the measurement sessions. Therefore, non-cooperative patients, such as infants, are recognized by the new image space integrating several sources of variability due to their free head pose and eye behavior.

In the last decade, deep learning (DL) has emerged as a powerful approach to computer vision tasks, in contrast to classical methods. This is because DL models could extract the necessary features from complex background images, resulting in significant success in the computer vision field. In this work, the problem of extending pupillometry to integrate very young patients is approached by exploiting DL techniques. This will be achieved by providing an end-to-end solution that automates the collection of PLR information from non-cooperative patients. First, Convolution Neural Network (CNN) is employed to reduce the image space complexity to the traditional eye-centered one. Second, to achieve accurate PLR measurements, a novel post-processing algorithm is proposed that utilizes depth information to define pupil size at the subpixel level. Third, a decision support tool for enhancing the objectivity of the measurements is being proposed. This tool provides valuable gaze information for guessing the accommodation reflex which is a major factor in altering the PLR objectivity toward the predefined light stimulus. This is achieved via a second model leveraging DL techniques.

It is worth noting that each mentioned process was tested on publically available datasets and exhibited a satisfying performance adequate for the case of use. The pupil

region extraction model achieved a Normalized Mean Error (NME) of around 4% on the 300w and the WFLW datasets.
The pupil size estimator was tested against three different datasets and performed an accuracy level of 76.04% and a precision level of 81.75% in the Swirski dataset.
In terms of the objectivity enhancement solution, the employed DENSNET achieved a mean error of 7.6° on the EVE dataset.

By integrating the aforementioned components into a pupillometry framework, increased flexibility in accommodating the patient's behavior could be achieved. The structure of this thesis will revolve around three key components: pupil detection, pupil size estimation, and objectivity enhancement. These components will be presented in an alternating manner, discussing their foundations in the literature in Chapter 1, the methods employed in Chapter 2, and the performance measurements in Chapter 4. Furthermore, Chapter 3 provides detailed information about the materials used in the implementation and evaluation processes. To provide readers with more comprehensive information, appendices containing additional material on various aspects discussed in the main body of the text are included in Chapter A.

List of Figures

List of Tables

Appendix A
Appendices

1 Optimization algorithm, ADAM

: Finding a local minimum for the cost function Equation (2.7) defined in Section 1.3 is performed iteratively. As a differentiable function, it is allowed to compute the gradient of the cost function and use it to predict the next iteration step, which is proportional to the opposite of the gradient direction. The process that enables us to converge toward a local minimum is called gradient descent GD. In the literature, one can find gradient descent variants that are more advanced than vanilla GD. ADAM, which stands for adaptive moment estimation was first introduced in 2015 by Kingma et al. [101]. It is visible that ADAM is the combination of stochastic gradient descent SGD Bottou [102] and the root mean square propagation RMS prop [103]. The following equations translate the iterative optimization process for ADAM. It is defined by W the ensemble of weights optimized during learning. The first-order momentum and the second-order un-centered momentum are respectively initialized as follows: $V_{dw} = 0, S_{dw} = 0$ On iteration t, Compute d_w the gradient on the current mini-batch of W using gradient descent then update the first order momentum V_{dw} as follows: $V_{dw} = \beta_1 V_{dw} + (1 - \beta_1) d_w$. Then, the following is executed to calculate the second order un-centered momentumsS_{dw}: $S_{dw} = \beta_2 S_{dw} + (1 - \beta_2)\, d_w^2$.

For statistical requirements, bias correction is added for both elements as follows: $V_{dw}^{corrected} = \frac{V_{dw}}{1-\beta_1^t}$ $S_{dw}^{corrected} = \frac{S_{dw}}{1-\beta_2^t}$ where β_1 and β_2 have intuitive interpretations and do not have a large range of tuning. Finally, the weight update is reached as follows:

$$W = W - \frac{\alpha V_{dw}^{corrected}}{\sqrt{S_{dw}^{corrected} + \epsilon}} \tag{A.1}$$

where α is the learning rate. The mentioned hyper-parameters are set as follows: $\alpha = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The detailed algorithm combines the effect of momentum GD with RMS prop-based GD. Adam is very efficient as it is straightforward to implement and requires only a small memory size besides being well suited for large data problems. Finally, Adam is well-fitted for training deep neural networks such as CNN.

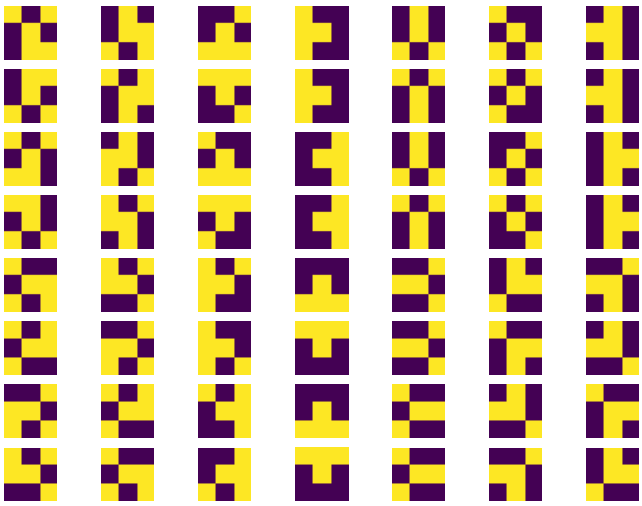


Figure A.1: Different kernels for junction detection where yellow pixels are on and purple pixels are off (Best viewed in color.)

2 Junction Detection

In Section 2.3 presented in Chapter 2, junctions were localized and omitted from detected edges. This was achieved via a multi-convolution process with already-defined kernels. In Figure A.1, the different kernels that were exploited for this operation are presented. Note that as explained in the referenced section, junctions were omitted from the detected edges to convert them into mono-branch edges which facilitates the pixel-wise operations on them.

3 Extremes Detection

In Section 2.3 presented in Chapter 2, extremes were localized in detected edges. This was achieved via a multi-convolution process with already-defined kernels. In Figure A.2, the different kernels that were exploited for this operation are presented. Note that as explained in the referenced section, extremes were detected in edges to convert to define heads and tails for them which facilitate pixel-wise operation, especially for the neighbor-based ordering of the edges.

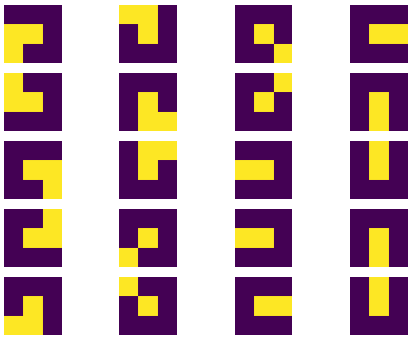


Figure A.2: Different kernels for extremes detection where yellow pixels are on and purple pixels are off (Best viewed in color.)

Bibliography

[1] Sungpyo Hong, Joanna Narkiewicz, and Randy H Kardon. “Comparison of pupil perimetry and visual perimetry in normal eyes: decibel sensitivity and variability”. In: *Investigative ophthalmology & visual science* 42.5 (2001), pp. 957–965.

[2] He Zhao et al. “Chromatic pupillometry isolation and evaluation of intrinsically photosensitive retinal ganglion cell-driven pupillary light response in patients with retinitis pigmentosa”. In: *Frontiers in Human Neuroscience* 17 (2023).

[3] Birgit Lorenz et al. “Early-onset severe rod–cone dystrophy in young children with RPE65 mutations”. In: *Investigative ophthalmology & visual science* 41.9 (2000), pp. 2735–2742.

[4] Stephen Russell et al. “Efficacy and safety of voretigene neparvovec (AAV2-hRPE65v2) in patients with RPE65-mediated inherited retinal dystrophy: a randomised, controlled, open-label, phase 3 trial”. In: *The Lancet* 390.10097 (2017), pp. 849–860.

[5] Samuel G Jacobson et al. “Defining the residual vision in leber congenital amaurosis caused by RPE65 mutations”. In: *Investigative ophthalmology & visual science* 50.5 (2009), pp. 2368–2375.

[6] Birgit Lorenz et al. “Chromatic pupillometry dissects function of the three different light-sensitive retinal cell populations in RPE65 deficiency”. In: *Investigative ophthalmology & visual science (IOVS)* 53.9 (2012), pp. 5641–5652.

[7] Neruban Kumaran, Anthony T Moore, Richard G Weleber, and Michel Michaelides. “Leber congenital amaurosis/early-onset severe retinal dystrophy: clinical features, molecular genetics and therapeutic interventions”. In: *British journal of ophthalmology* 101.9 (2017), pp. 1147–1154.

[8] Jackson Beatty. “Task-evoked pupillary responses, processing load, and the structure of processing resources.” In: *Psychological bulletin* 91.2 (1982), p. 276.

[9] Fabian Timm and Erhardt Barth. “Accurate eye centre localisation by means of gradients.” In: *Visapp* 11 (2011), pp. 125–130.

[10] Lech Świrski, Andreas Bulling, and Neil Dodgson. “Robust real-time pupil tracking in highly off-axis images”. In: *Proceedings of the symposium on eye tracking research and applications*. 2012, pp. 173–176.

[11] Babak Zandi et al. “PupilEXT: Flexible open-source platform for high-resolution pupillometry in vision research”. In: *Frontiers in neuroscience* (2021), p. 603.

[12] Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. “PuRe: Robust pupil detection for real-time pervasive eye tracking”. In: *Computer Vision and Image Understanding* 170 (2018), pp. 40–50.

[13] Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. “PuReST: Robust pupil tracking for real-time pervasive eye tracking”. In: *Proceedings of the 2018 ACM symposium on eye tracking research & applications*. 2018, pp. 1–5.

[14] Barnabás Takács and Harry Wechsler. “Detection of faces and facial landmarks using iconic filter banks”. In: *Pattern Recognit.* 30.10 (1997), pp. 1623–1636.

[15] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. “Active appearance models”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 23.6 (2001), pp. 681–685.

[16] Marcin Kopaczka, Kemal Acar, and Dorit Merhof. “Robust Facial Landmark Detection and Face Tracking in Thermal Infrared Images using Active Appearance Models.” In: *VISIGRAPP (4: VISAPP)*. 2016, pp. 150–158.

[17] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. “Active shape models-their training and application”. In: *Comput Vis Image Underst* 61.1 (1995), pp. 38–59.

[18] Ting-Chia Hsu, Yea-Shuan Huang, and Fang-Hsuan Cheng. “A novel ASM-based two-stage facial landmark detection method”. In: *Pacific-Rim Conference on Multimedia (PCM)*. 2010, pp. 526–537.

[19] Yue Wu et al. “Facial landmark detection with tweaked convolutional neural networks”. In: *IEEE PAMI* 40.12 (2017), pp. 3067–3074.

[20] Daniel Merget, Matthias Rock, and Gerhard Rigoll. “Robust facial landmark detection via a fully-convolutional local-global context network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2018, pp. 781–790.

[21] Khalil Khan et al. “A multi-task framework for facial attributes classification through end-to-end face parsing and deep convolutional neural networks”. In: *Sensors* 20.2 (2020), p. 328.

[22] Jiankang Deng, Qingshan Liu, Jing Yang, and Dacheng Tao. “M3 csr: Multi-view, multi-scale and multi-component cascade shape regression”. In: *Image Vis. Comput.* 47 (2016), pp. 19–26.

[23] Qingshan Liu, Jing Yang, Jiankang Deng, and Kaihua Zhang. “Robust facial landmark tracking via cascade regression”. In: *Pattern Recognit.* 66 (2017), pp. 53–62.

[24] Daniela O Medley, Carlos Santiago, and Jacinto C Nascimento. "Deep active shape model for robust object fitting". In: *IEEE Trans. Image Process.* 29 (2019), pp. 2380–2394.

[25] Lisha Chen, Hui Su, and Qiang Ji. "Deep structured prediction for facial landmark detection". In: *Adv Neural Inf Process Syst.* 32 (2019).

[26] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. "Joint training of a convolutional network and a graphical model for human pose estimation". In: *Adv Neural Inf Process Syst.* 27 (2014).

[27] Joe Yue-Hei Ng, Fan Yang, and Larry S Davis. "Exploiting local features from deep networks for image retrieval". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 53–61.

[28] Yi Sun, Xiaogang Wang, and Xiaoou Tang. "Deep convolutional network cascade for facial point detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2013, pp. 3476–3483.

[29] Xi Chen et al. "Delving deep into coarse-to-fine framework for facial landmark localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 142–149.

[30] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. "Facial landmark detection by deep multi-task learning". In: *European conference on computer vision (ECCV)*. Springer. 2014, pp. 94–108.

[31] Zhenliang He et al. "A fully end-to-end cascaded cnn for facial landmark detection". In: *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. 2017, pp. 200–207.

[32] Ivan Gogić, Jörgen Ahlberg, and Igor S Pandžić. "Regression-based methods for face alignment: A survey". In: *IEEE Signal Process. Mag.* 178 (2021), p. 107755.

[33] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. "Striving for simplicity: The all convolutional net". In: *arXiv preprint arXiv:1412.6806* (2014).

[34] Rachida Hannane, Abdessamad Elboushaki, and Karim Afdel. "A divide-and-conquer strategy for facial landmark detection using dual-task CNN architecture". In: *Pattern Recognition* 107 (2020), p. 107504.

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2015, pp. 3431–3440.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*. Springer. 2015, pp. 234–241.

[37] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation". In: *European conference on computer vision (ECCV)*. Springer. 2016, pp. 483–499.

[38] Adrian Bulat and Georgios Tzimiropoulos. "Human pose estimation via convolutional part heatmap regression". In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 717–732.

[39] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. "Why does unsupervised pre-training help deep learning?" In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics (AISTATS)*. JMLR Workshop and Conference Proceedings. 2010, pp. 201–208.

[40] Dilip K Prasad, Maylor KH Leung, and Siu-Yeung Cho. "Edge curvature and convexity based ellipse detection method". In: *Pattern Recognition* 45.9 (2012), pp. 3204–3221.

[41] Peter D Kovesi. *MATLAB and Octave functions for computer vision and image processing*. 2000.

[42] Dilip K Prasad, Chai Quek, Maylor KH Leung, and Siu-Yeung Cho. "A parameter independent line fitting method". In: *The First Asian Conference on Pattern Recognition*. IEEE. 2011, pp. 441–445.

[43] Zepeng Wang, Derong Chen, Jiulu Gong, and Changyuan Wang. "Fast high-precision ellipse detection method". In: *Pattern Recognition* 111 (2021), p. 107741.

[44] C-H Teh and Roland T. Chin. "On the detection of dominant points on digital curves". In: *IEEE Transactions on pattern analysis and machine intelligence* 11.8 (1989), pp. 859–872.

[45] Dongheng Li, David Winfield, and Derrick J Parkhurst. "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*. IEEE. 2005, pp. 79–79.

[46] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. "Deep learning for monocular depth estimation: A review". In: *Neurocomputing* 438 (2021), pp. 14–33.

[47] Robert R Henderson, Margaret M Bradley, and Peter J Lang. "Emotional imagery and pupil diameter". In: *Psychophysiology* 55.6 (2018), e13050.

[48] Adam Fisch. "Clinical examination of the cranial nerves". In: *Nerves and Nerve Injuries*. Elsevier, 2015, pp. 195–225.

[49] Sanjeev Kasthurirangan and Adrian Glasser. "Characteristics of pupil responses during far-to-near and near-to-far accommodation". In: *Ophthalmic and Physiological Optics* 25.4 (2005), pp. 328–339.

[50] Hanyang Yu et al. "Is Ocular Accommodation Influenced by Dynamic Ambient Illumination and Pupil Size?" In: *International Journal of Environmental Research and Public Health* 19.17 (2022), p. 10490.

[51] Michael S Franklin et al. *Window to the wandering mind: Pupillometry of spontaneous thought while reading*. 2013.

[52] Yong-Goo Shin, Kang-A Choi, Sung-Tae Kim, and Sung-Jea Ko. "A novel single IR light based gaze estimation method using virtual glints". In: *IEEE Transactions on Consumer Electronics* 61.2 (2015), pp. 254–260.

[53] Mark F Bradshaw, Andrew D Parton, and Andrew Glennerster. "The task-dependent use of binocular disparity and motion parallax information". In: *Vision research* 40.27 (2000), pp. 3725–3734.

[54] Dan Witzner Hansen and Qiang Ji. "In the eye of the beholder: A survey of models for eyes and gaze". In: *IEEE transactions on pattern analysis and machine intelligence* 32.3 (2009), pp. 478–500.

[55] Dario Cazzato, Marco Leo, Cosimo Distante, and Holger Voos. "When I look into your eyes: A survey on computer vision contributions for human gaze estimation and tracking". In: *Sensors* 20.13 (2020), p. 3739.

[56] Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato, and Hideki Koike. "An incremental learning method for unconstrained gaze estimation". In: *European conference on computer vision*. Springer. 2008, pp. 656–667.

[57] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. "Appearance-based gaze estimation in the wild". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4511–4520.

[58] Kyle Krafka et al. "Eye tracking for everyone". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2176–2184.

[59] Kenneth Alberto Funes Mora and Jean-Marc Odobez. "Person independent 3d gaze estimation from remote rgb-d cameras". In: *2013 IEEE International Conference on Image Processing*. IEEE. 2013, pp. 2787–2791.

[60] Seonwook Park, Adrian Spurr, and Otmar Hilliges. "Deep pictorial gaze estimation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 721–738.

[61] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. "Appearance-based gaze estimation with deep learning: A review and benchmark". In: *arXiv preprint arXiv:2104.12668* (2021).

[62] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. "Learning-by-synthesis for appearance-based 3d gaze estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1821–1828.

[63] Jimmy Ren et al. "Accurate single stage detector using recurrent rolling convolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2017, pp. 5420–5428.

[64] Nanne Van Noord and Eric Postma. "Learning scale-variant and scale-invariant features for deep image classification". In: *Pattern Recognit.* 61 (2017), pp. 583–592.

[65] Yichong Xu et al. “Scale-invariant convolutional neural networks”. In: *arXiv preprint arXiv:1411.6369* (2014).

[66] Seung-Wook Kim et al. “Parallel feature pyramid network for object detection”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 234–250.

[67] Arjun Jain et al. “Learning human pose estimation features with convolutional networks”. In: *arXiv preprint arXiv:1312.7302* (2013).

[68] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Efficient belief propagation for early vision”. In: *Int. J. Comput. Vis.* 70.1 (2006), pp. 41–54.

[69] Xinyao Wang, Liefeng Bo, and Li Fuxin. “Adaptive wing loss for robust face alignment via heatmap regression”. In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*. 2019, pp. 6971–6981.

[70] Thanh Phuong Nguyen and Isabelle Debled-Rennesson. “A discrete geometry approach for dominant point detection”. In: *Pattern Recognition* 44.1 (2011), pp. 32–44.

[71] Michele Fornaciari, Andrea Prati, and Rita Cucchiara. “A fast and effective ellipse detector for embedded vision applications”. In: *Pattern Recognition* 47.11 (2014), pp. 3693–3708.

[72] David H. Eberly. “Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid”. In: 2006.

[73] Zhengyou Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000), pp. 1330–1334.

[74] Pascal Monasse, Jean-Michel Morel, and Zhongwei Tang. “Three-step image rectification”. In: *BMVC 2010-British Machine Vision Conference*. BMVA Press. 2010, pp. 89–1.

[75] Adina T Michael-Titus, Patricia Revest, and Peter Shortland. *The Nervous System, Systems of the Body Series*. Elsevier Health Sciences, 2010. Chap. 6, p. 116.

[76] Sheng-Wen Shih, Yu-Te Wu, and Jin Liu. “A calibration-free gaze tracking technique”. In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. Vol. 4. IEEE. 2000, pp. 201–204.

[77] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. “Mpiigaze: Real-world dataset and deep appearance-based gaze estimation”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.1 (2017), pp. 162–175.

[78] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. “It’s written all over your face: Full-face appearance-based gaze estimation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 51–60.

[79] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[80] Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. "Towards End-to-end Video-based Eye-Tracking". In: *European Conference on Computer Vision (ECCV)*. 2020.

[81] Adam Paszke et al. "Automatic differentiation in PyTorch". In: (2017).

[82] Wes McKinney et al. "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.

[83] Christos Sagonas et al. "300 faces in-the-wild challenge: Database and results". In: *Image Vis Comput.* 47 (2016), pp. 3–18.

[84] Vuong Le et al. "Interactive facial feature localization". In: *European conference on computer vision (ECCV)*. 2012, pp. 679–692.

[85] Wayne Wu et al. "Look at boundary: A boundary-aware face alignment algorithm". In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2018, pp. 2129–2138.

[86] Keshav Seshadri and Marios Savvides. "Robust modified active shape model for automatic facial landmark annotation of frontal faces". In: *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*. 2009, pp. 1–8.

[87] Stephen Milborrow and Fred Nicolls. "Locating facial features with an extended active shape model". In: *European conference on computer vision (ECCV)*. 2008, pp. 504–513.

[88] Yi Yang and Deva Ramanan. "Articulated human detection with flexible mixtures of parts". In: *IEEE PAMI* 35.12 (2012), pp. 2878–2890.

[89] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[90] Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. "Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments". In: *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications*. 2016, pp. 139–142.

[91] Wolfgang Fuhl et al. "Excuse: Robust pupil detection in real-world scenarios". In: *Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015 Proceedings, Part I 16*. Springer. 2015, pp. 39–51.

[92] Ahmed Gdoura, Markus Degünther, Birgit Lorenz, and Alexander Effland. "Combining CNNs and Markov-like Models for Facial Landmark Detection with Spatial Consistency Estimates". In: *Journal of Imaging* 9.5 (2023), p. 104.

[93] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. "2d human pose estimation: New benchmark and state of the art analysis". In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 3686–3693.

[94] Hui Li et al. "Towards Accurate Facial Landmark Detection via Cascaded Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 4176–4185.

[95] Wenyan Wu and Shuo Yang. "Leveraging intra and inter-dataset variations for robust face alignment". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 150–159.

[96] Xiaoqian Yue et al. "Multi-task adversarial autoencoder network for face alignment in the wild". In: *Neurocomputing* 437 (2021), pp. 261–273.

[97] Meilu Zhu, Daming Shi, Mingjie Zheng, and Muhammad Sadiq. "Robust facial landmark detection via occlusion-adaptive deep networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3486–3496.

[98] Xu Zou et al. "Learning robust facial landmark detection via hierarchical structured ensemble". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 141–150.

[99] Haibo Jin, Shengcai Liao, and Ling Shao. "Pixel-in-pixel net: Towards efficient facial landmark detection in the wild". In: *Int. J. Comput. Vis.* 129.12 (2021), pp. 3174–3194.

[100] Amir Zadeh, Yao Chong Lim, Tadas Baltrusaitis, and Louis-Philippe Morency. "Convolutional experts constrained local model for 3d facial landmark detection". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 2519–2528.

[101] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[102] Léon Bottou. "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer. 2010, pp. 177–186.

[103] Tijmen Tieleman and G Hinton. "Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning". In: *Technical report* (2017).

List of Publications

Ahmed Gdoura, Markus Degünther, Birgit Lorenz, and Alexander Effland. “Combining CNNs and Markov-like Models for Facial Landmark Detection with Spatial Consistency Estimates”. In: Journal of Imaging 9.5 (2023), p. 104.

Eidesstattliche Erklärung

„Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne unzulässige Hilfe oder Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder nichtveröffentlichten Schriften entnommen sind, und alle Angaben, die auf mündlichen Auskünften beruhen, sind als solche kenntlich gemacht. Bei den von mir durchgeführten und in der Dissertation erwähnten Untersuchungen habe ich die Grundsätze guter wissenschaftlicher Praxis, wie sie in der „Satzung der Justus-Liebig-Universität Gießen zur Sicherung guter wissenschaftlicher Praxis“ niedergelegt sind, eingehalten sowie ethische, datenschutzrechtliche und tierschutzrechtliche Grundsätze befolgt. Ich versichere, dass Dritte von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen, und dass die vorgelegte Arbeit weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde zum Zweck einer Promotion oder eines anderen Prüfungsverfahrens vorgelegt wurde. Alles aus anderen Quellen und von anderen Personen übernommene Material, das in der Arbeit verwendet wurde oder auf das direkt Bezug genommen wird, wurde als solches kenntlich gemacht. Insbesondere wurden alle Personen genannt, die direkt und indirekt an der Entstehung der vorliegenden Arbeit beteiligt waren. Mit der Überprüfung meiner Arbeit durch eine Plagiatserkennungssoftware bzw. ein internetbasiertes Softwareprogramm erkläre ich mich einverstanden.“

Ort/Datum	Unterschrift