Universitat Politècnica de Catalunya
Universitat de Barcelona
Universitat Rovira i Virgili

Master Degree in Artificial Intelligence

Master thesis

Spatio-temporal Gaze Estimation for Human-Machine and Human-Human Interaction

Authors:
Alberto López Sánchez

Defense date
31 de Enero del 2020

Project leaders
Sergio Escalera
Cristina Palmero

Acknowledgements

Special thanks to my friend Mr.Javier López Calderón to let me borrow hardware to build a computer without which I could not have performed all the necessary experiments to complete this project, also thanks for listening in the difficult moments and give me advice when needed.

Thanks to my directors Dr.Sergio Escalera and Cristina Palmero, for driving the project and guide me in this master thesis.

Abstract

Eye gaze is an important non-verbal cue in human-human and human-machine interactions. In this master thesis, we explore optical flow as a new feature of temporal information added to face and eyes to perform 3D gaze estimation from remote cameras in a mid-distance scenario. We propose new models that combine face, optical flow from the face between the last two frames, eyes, and face landmarks as individual streams in a CNN to estimate gaze using the last two images. We also develop a recurrent model that exploits the dynamic nature of gaze by feeding the learned features of all the frames in a sequence to a many-to-one recurrent module that predicts the 3D gaze vector of the last frame. Our experiments show that, with the addition of the temporal information of optical flow, static models can perform just as well as recurring models, while maintaining lower complexity and faster inference than recurrent ones.

keywords: python, deep, machine, learning, optical flow, recurrent, lstm, gru, eyediap, dataset, eye, gaze, prediction, keras, ablation, median, filter

Contents

List of Tables

List of Figures

1 Introduction

It has been shown that gaze is an essential component in non-verbal communication reflecting all kinds of information about what a human being thinks or feels, both in conversations with other people, and when we are interacting with a machine [17, 10]. This is why the research community is focusing on being able to obtain the gaze vector using only conventional hardware that you can easily find as an RGB camera on a mobile phone and in realistic conditions (different angles, lighting conditions, image quality, person variability) without the need for calibration or setting up a dedicated environment for it.

This master thesis focus on the regression of gaze vectors in mid-distance scenario for human-human and human-computer interaction purposes. We want to exploit visual information to define non-invasive technology able to regress accurate gaze vectors without the need for hardware-specific eye trackers. In particular, we are interested in exploiting the spatio-temporal nature of eye dynamics to improve the Deep regression results of gaze estimation.

Previous work in the field just explored the appearance nature of eyes and regressing the vectors that better approximate the appearance information from still images. [22] Given the natural movement of the eyes, in 2018, Palmero et al. published a work showing that recurrent models can further exploit gaze information of consecutive frames and further improve gaze regression. Here we are interested in modeling another kind of motion pattern from eyes to see up to which degree spatio-temporal information can be exploited to further boost the performance of gaze regression.

In particular, we focus on the analysis of optical flow, since it defines the local movement of consecutive images regions, defining a compact representation of motion. Optical flow has shown to be a powerful representation in action and gesture recognition in the computer vision community [6], contrary to 3DCNNs, that capture spatio-temporal information by increasing largely the number of parameters and are prone to overfitting [26]. Optical flow images, instead, provide a compact representation where standard 2D kernels can directly learn motion information. so, our goals are:

- To provide additional motion image representation based on optical flow to classic eye and face appearance image regions.
- To analyze the effect of optical flow representation, enhancing recognition performance in image sequences of gaze regression.
- To further combine the findings with other classical temporal exploitation techniques, such as temporal post-processing and recurrent architectures.

2 Related work

Methods to estimate gaze are categorized as model-based or appearance-based.

Model-based approaches require high-resolution images or calibration specific parameters to estimate personal eye parameters. [19, 27, 28, 31, 32]. On the other hand, appearance-based models can be used in low-resolution images or a mid-distance scenario, as these methods learn a direct mapping from intensity images or extracted eye features to gaze directions. [2, 18, 25]

The two main issues the researchers have to face are appearance variations and head pose variations. In the work of Fischer *et al.* [7], they consider the issue of reliable gaze estimation in natural environments. So large camera-to-subject distances and high head pose and eye gaze angle variations are expected. They recorded a dataset with videos of varied gaze and head images in a natural environment with a motion capture system and eye-tracking glasses. The eye-tracking glasses are a problem because that makes the appearance of the faces in the videos not clean. To be able to estimate the gaze in persons without the eye-tracker, they inpaint this area using a Generative adversarial network (GANs) [9] to try to recreate the original face image. They also present a new real-time algorithm involving appearance-based deep convolutional neural networks that they try in several datasets included their own. We decided not to select this dataset because the regenerated images by the GAN has artifacts.

2.1 Dynamic gaze estimation

Kellnhofer *et al.* [13] creates a dataset with 238 subjects with the objective to be able to estimate the gaze in indoor and outdoor environments. They intend to make a temporal model able to estimate the gaze of the people even when they are not looking towards the camera.

Wang *et al.* [29] propose to exploit the dynamics of eye movement. Studies show that there are several common types of eye movement, regardless of content and topics, such as fixation, saccade, and smooth pursuits. Therefore, adding generic dynamics of eye movement will improve generalization capabilities. In particular, they propose a probabilistic graphical model named Dynamic Gaze Transition Network (DGTN), to capture the dynamics of the underlying eye movement. Wich is used to refining the predictions already done by other static methods. They also develop a new dataset to estimate screen targets.

Finally, we analyze the work of Palmero *et al.* [22]. They propose to use a multi-modal recurrent convolutional neural network (CNN). They combine face, eyes region, and face landmarks as individual streams in a CNN to estimate gaze in still images from remote cameras. Also, they propose a many-to-one recurrent network that predicts the 3D gaze vector of the last frame, achieving a significant improvement of 14.6% over state of the art on the EYEDIAP dataset (see Section 3), further improved by 4% when the temporal modality is included. We decided to use their work as a baseline and add the optical flow information in their models.

2.2 Ways to use temporal info

In the work of Feichtenhofer *et al.* [6] they fuse appearance with spatio-temporal information for action recognition in videos using Convolutional Neural Networks. They propose a ConvNet architecture for spatio-temporal fusion of video snippets. The network they propose is similar to the networks we develop in this work. They use a stack of the optical flow of the frames of the video as spatio-temporal information as the second input of the network.

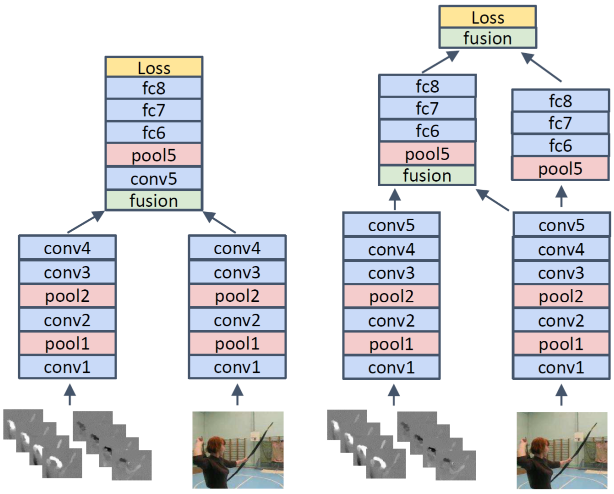


Figure 1: The networks proposed in the paper Convolutional Two-Stream Network Fusion for Video Action Recognition, showing how to combine the two streams in two possible ways. Image extracted from [6].

In this work, we analyze several multi-stream CNN network for person and head pose-independent 3D gaze estimation for a mid-distance scenario, adding optical flow for temporal information as a key component. To the best of our knowledge, this is the first work using optical flow for gaze estimation.

3 Dataset

The dataset we use for this project is the EYEDIAP dataset [8]. We choose the EYEDIAP dataset because it is the only one that contains image sequences without visual artifacts in the face (for instance, the RT-GENE [7] has this problem). This dataset intends to fill the need for a standard database for gaze estimation from remote RGB, and RGB-D (standard vision and depth) cameras. It has a total of 96 recording sessions, each with different characteristics. The RGB stream has two versions, the VGA resolution (640x480 pixels) at 30 frames per second and the High definition resolution. For this work, we chose to use the VGA resolution to mimic mid-distance scenarios. The subjects recorded in the sessions were 16 adults, 12 male and four female participants with varied age range.

The recording methodology was designed to systematically include and isolate most of the variables which affect the remote gaze estimation algorithms, such as head pose variations, person variation, changes in environment and sensing condition. In the Figure 2 we can see the recording Setup.

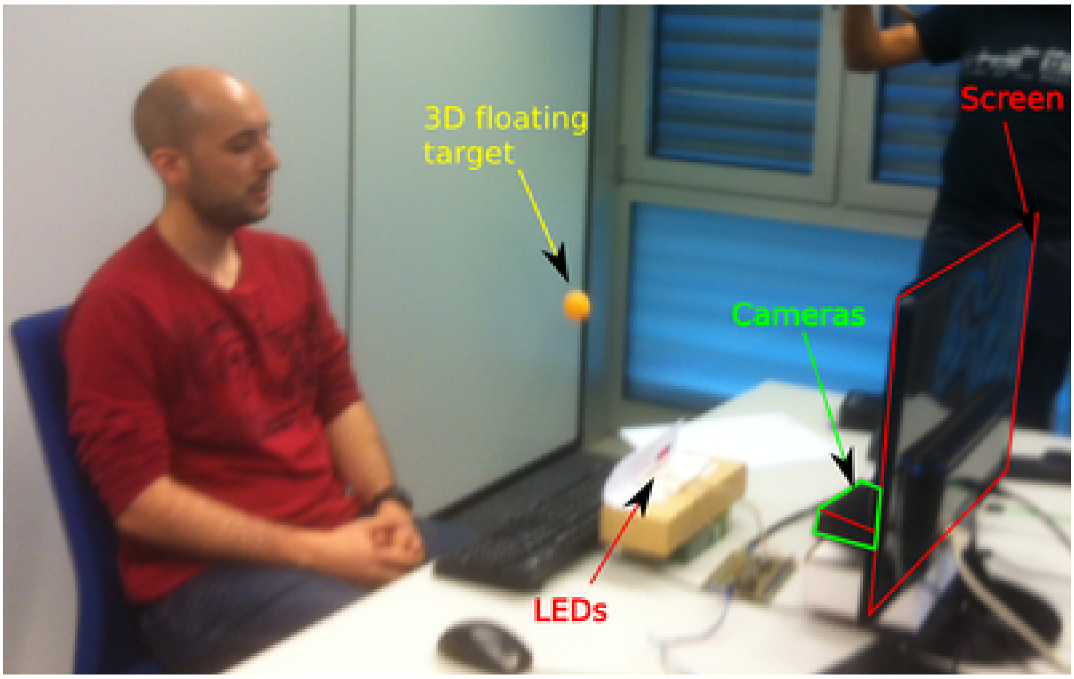


Figure 2: Setup of the recording room for the EYEDIAP dataset where can be seen the subject, the floating target, the screen target and the cameras. (from [8])

The sessions in the dataset have an average of 4538 frames recorded and were done in two different illumination conditions (A and B), and can be split into three types:

1. Discrete screen target (DS) a small circle appears at the screen every 1.1 seconds at random locations.

2. Continuous screen target (CS) the circle appears at the screen and moves along a random trajectory for 2 seconds.

3. 3D floating target (FT): A small 4cm diameter ball was moved within a 3d region in front of the subject. The distance between the camera and the participant was 1.2 meters in this case.

The floating target recorded sessions consisted of two types:

1. The subject performs head movements (rotations and translations) in all directions while looking at the target, to obtain appearances of faces with very varied poses.

2. Fixed head while the subject looks at the objective, to get images where only changes the appearance of the eyes.

We decided to select the 66 sessions that consisted of looking at floating targets since we did not have time to test to compute all the experiments with all the dataset and in the related work we observed that the mid-range problem is what is more challenging.

4 Methodology

We present a 3D gaze regression method based on appearance, shape cues, and optical flow for still images and image sequences. First, we explain how we obtain the features for the models and formulate the problem. Then, we present the models designed for the experiments.

4.1 Gaze regression

The 3D gaze unit vector in the Camera Coordinate System (CCS) is represented as $\mathbf{g} = [g_x, g_y, g_z]^T \in \mathbb{R}^3$. This coordinate system has the origin in the central point between eyeball centers. Then with a calibrated camera, and knowing the head transformation (position and rotation), we can estimate $\mathbf{g}$ from a still image in the case of the static models and a from a sequence of images $\left\{(\mathbf{I}^{(i-4)}, \mathbf{I}^{(i-3)}, ..., \mathbf{I}^{\mathbf{i}}) | \mathbf{I} \in \mathbb{R}^{W \times H \times 3}\right\}$ in the recurrent models.

Wollaston [21] demonstrates that gaze direction could not be based only on the estimation where the irises are located within the lid aperture but also the rotation of the head is also important, see Figure 3. So we can conclude that eye images are not enough to estimate gaze direction, and the whole face image is needed.

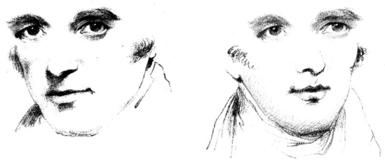


Figure 3: Same eyes in two different faces illustrating the Wollaston effect (From "On the Apparent Direction of Eyes in a Portrait," by W. H. Wollaston, 1824, Philosophical Transactions of the Royal Society of London, 114, p. 256. In the public domain)

In our approach, we jointly model appearance, geometry, and shape cues as a whole using the face image, a higher resolution of the eyes (scaled), the face landmarks, and the optical flow between the previous frame and the current one.

4.2 Features

In this section, we are going to explain how the features used by the models are extracted.

4.2.1 Landmarks

Facial landmarks can be used as global shape cues to encode spatial relationships and geometric constraints. Current state-of-the-art face alignment approaches are robust enough to handle large appearance variability, extreme head poses, and occlusions, being especially useful when the dataset used for gaze estimation does not contain such variability. Facial landmarks are mainly correlated with head orientation, eye position, eyelid openness, and eyebrow movement, which are valuable features for our task.

The 3D landmarks are extracted using the state-of-the-art method of Bulat and Tzimiropoulos [4], which is based on stacked hourglass networks [20] (see Figure 4). This method extracts landmarks in 3D coordinates.

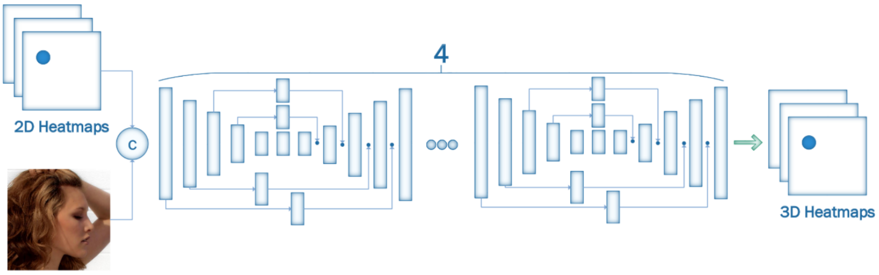
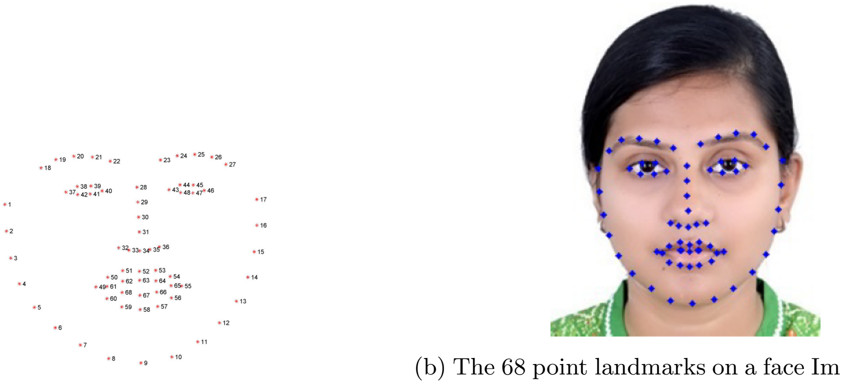


Figure 4: Image extracted from [4]. This network takes as input the RGB image and the 2D landmarks and outputs the corresponding 3D projections of the 2D landmarks.

For each frame of the dataset a 68-landmark vectors denoted by $\mathbf{L} = \left\{ (l_x, l_y, l_z)_c \, | \forall c \in [1, \ldots, 68] \right\}$ is extracted. An example can be seen in the Figure 5



(a) 68 points mark-up used for face annotations.

(b) The 68 point landmarks on a face Image. Image extracted from [12]

Figure 5: Visualization of the coverage of the 68 face landmarks.

4.2.2 Eyes region extraction

The eyes region has been obtained cropping the eyes using the face landmarks. The rectangle region of each eye is obtained by using the landmarks corresponding to each eye and making it a % bigger, so the eye is completely captured even if the landmark is not accurate enough.

Due to dealing with wide head pose ranges, some eye images may not depict the whole eye, containing mostly background or other surrounding facial parts instead. We use a single image composed of two patches of centered left and right eyes to reduce this problem, so at least one of the eyes will contain all the information needed.

A sample of the input of the eyes region is shown in Figure 6

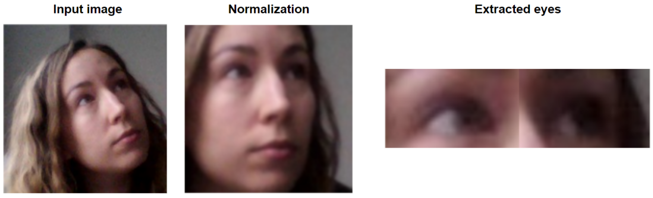


Figure 6: An example of the original face image, the normalized face and the eyes region extracted. Face images extracted from [22].

4.2.3 Optical flow

Optical flow is the motion of objects between consecutive frames of a sequence, caused by the relative movement between the object and camera and has several uses, including video encoding, video re-timing, and video stabilization [30]. However, in this work, we are going to use it to track the movement of the face and eyes between two frames. In the Figure 7 an example of applying it to two consecutive frames could be seen.

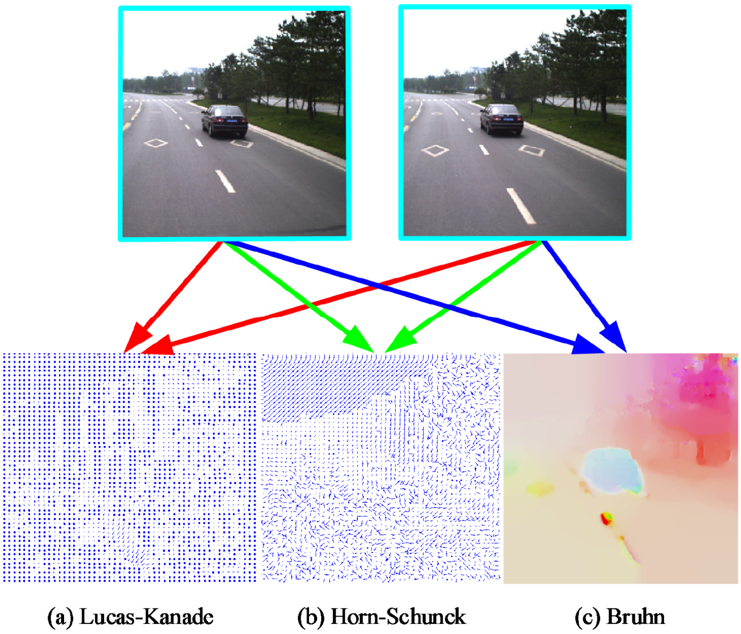


Figure 7: Example of resulting displacement vectors using several optical flow methods.

It can be expressed as can be shown in the Figure 8:

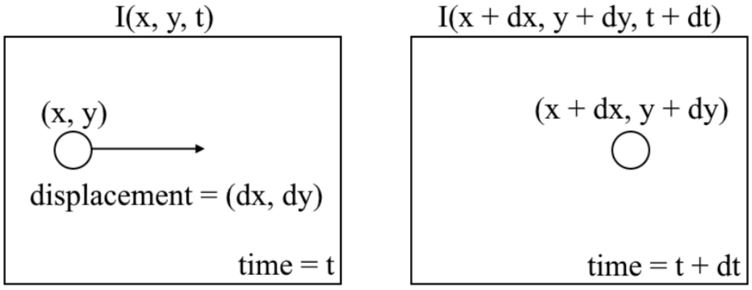


Figure 8: Graphical representation of the optical flow problem.

where I is a function of (x, y) space and (t) time that give us the intensity of a pixel in the coordinates x, y in the frame t. The displacement (dx, dy) represents how much a pixel has been moved in x,y.

We want an algorithm able to compute how much a pixel has been moved, which means we need an algorithm that computes dx and dy. For this, we have to solve the next equation:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{1}$$

Using the Taylor Series Approximation in the right-hand side of equation and removing common terms:

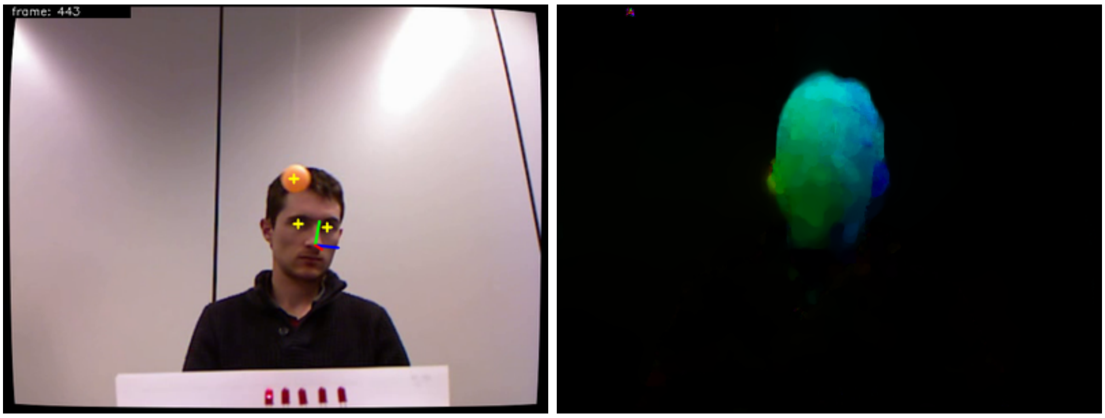
$$\begin{aligned} &I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + \dots \\ &\Rightarrow \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0 \end{aligned} \tag{2}$$

And finally dividing by dt to derive the optical flow equation:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{3}$$

where $u = dx/dt$ and $v = dy/dt$.

We conclude that we need to solve $u(dx/dt)$ and $v(dy/dt)$ to determine the movement over time. This equation has two unknowns and cannot be solved without adding some other restrictions. This is known as the aperture problem of the optical flow algorithms. To find the optical flow, we need another set of equations, with some additional constraints. Optical flow methods introduce additional conditions for estimating the actual flow. We choose the method from C. Liu. [15] to solve these equations, is a method based in the Lucas/Kanade [3] that combines Local and Global Optical flow methods and uses the RGB information of the image instead of only the intensity of the pixels usually obtained converting the images to greyscale. We choose this one instead of the classical Lucas/Kanade because, empirically, in the tests we did, we observed less noise and more precision in the results.



(a) Example of image I_t with floating target and moving head.

(b) Optical flow of the motion from frames I_{t-1} to I_t

Figure 9: Example of visualization of optical flow between two frames of the dataset

The optical flow is converted to RGB for visualization interpreting the dx and dy of each pixel as a vector converted to polar coordinates and using the angle as the color selector and the magnitude as the value in HSV color space.

4.3 Data normalization

A normalization step in 3D space and the 2D image is needed to be able to apply the gaze estimation model regardless of the original camera configuration. We use the normalization method of Sugano *et al.* [24]. The resulting image is a cropped image patch of size $W_n \times H_n$ centered in p showing only the head, where the head roll rotation has been removed. This normalization is achieved by applying a perspective warping transformation to the input image. An example can be seen in Figure 6.

The 3D gaze vector is unit normalized. Then, $\mathbf{g}_n$ (gaze vector normalized) is transformed to spherical coordinates (θ, ϕ), where θ and ϕ denote the horizontal and vertical direction angles, respectively. These two angles are what our regression model estimates.

The 2D angle representation is delimited in the range $[-\pi/2, \pi/2]$ and is computed as $\boldsymbol{\theta} = \arctan(g_x/g_z)$ and $\phi = \arcsin(-g_y)$, such that (0,0) represents looking straight ahead to the CCS origin.

The landmarks and optical flow are also normalized using the same warping matrix like the image face. The eye region is extracted from the already normalized face using the normalized landmarks, so no further normalization step is needed for the eyes.

The magnitudes of optical flow are also normalized before being used as a feature for the model. The two channels (dx, dy) are normalized individually using the following algorithm:

```
# First we find the maximum magnitude of the dx and dy in all the frames.
for frame in opticalflow_frames:
    normlize_dx = max(abs(frame[dx]))
    normlize_dy = max(abs(frame[dy]))
```

```
# Then we divide each frame by this maximum magnitudes.
for frame in opticalflow_frames:
    frame[dx] = frame[dx] / normalize_dx
    frame[dy] = frame[dx] / normalize_dy
```

4.4 Models

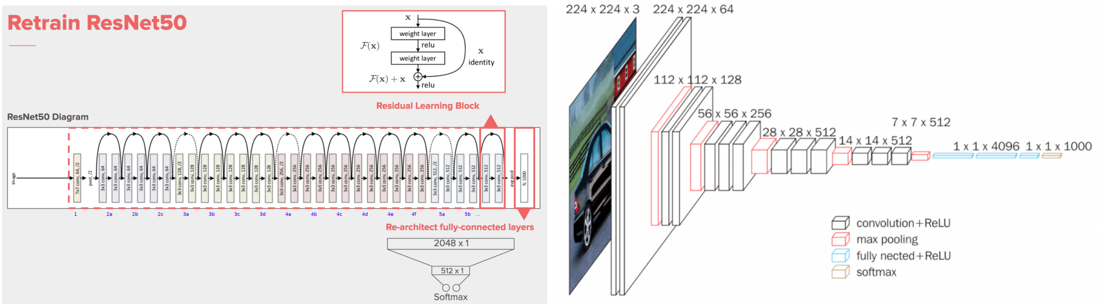
The models developed to solve the problem of regression of the estimation of the gaze are convolutional neural networks (CNN) combined with fully connected (FC) and recurrent layers.

In particular, we develop and train five new models that use as base models the VGG16 network [23] and the ResNet50 [11] network. Both are outstanding networks in the competitions of Large Scale Visual Recognition Challenge 2017 (ILSVRC2017) but in this case trained to do face recognition. Although it is not the task that we want to solve, they are useful for computing the embeddings of some of the features of our model, in particular, face, eyes, and optical flow of both face and eyes. Face recognition is a technology capable of identifying or verifying a person from a digital image. We expect that the embeddings will contain enough information to model all the aspects of the appearance of the face.

These backbones have the following characteristics:

VGG16 (see Figure 10 (b)) has a total of 138 million parameters and a depth of 16 layers. All the convolutional kernels are of size 3x3, and max pool kernels are of size 2x2 with a stride of two.

ResNet (see Figure 10 (a)) introduced residual blocks that have connections between layers, meaning that the output of a layer is a convolution of its input plus its input, avoiding the vanishing gradient problem allowing a much more deep network and reducing the number of total parameters. In particular, ResNet50 has 50 layers and 25.6 million parameters. An ensemble of these residual nets achieves a 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task.



(a) Resnet50 Diagram architecture.

(b) VGG16 Diagram architecture.

Figure 10: Networks used as base models

4.4.1 Static models

In the following sections, we are going to describe the models developed during this work and explain the particularities of each one and in what experiments are used. All the models have the number of neurons that defines the last fully connected layers already pre-defined as a sum of the neurons associated with the input streams in the following way:

- Face (Normalized or not) (224x224x3) RGB pixels: 4096 neurons
- Eyes (120x48x3) RGB pixels: 1536 neurons
- Optical flow of the face (224x224x3) (dx, dy, dy): 1536 neurons
- Landmarks (3D points [x,y,z]x68): 204 neurons

The number of neurons is decreased proportionally in the case of the eyes. Optical flow fewer neurons than the face because it codifies much less information than the face itself.

We divided the models depending on how many streams are used as input.

One stream The most basic network model consists only of the input of a face in RGB (224,224,3) pixels. The experiment that uses this model is (Normalized Face, 4096) that can be seen in Section 5.6.1. This model was also used to compute this same experiment but without normalizing the face.

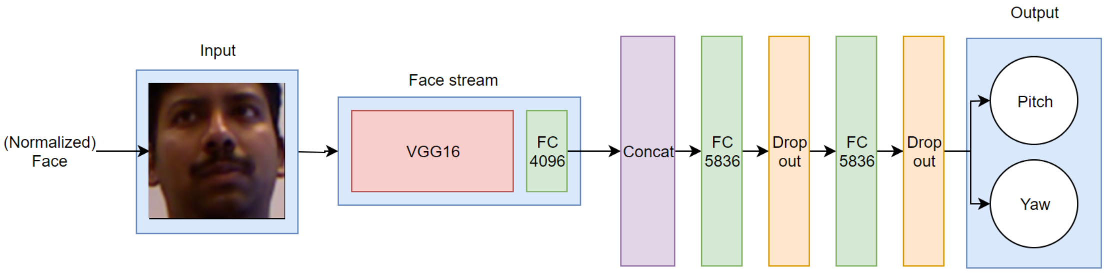


Figure 11: One stream model.

Two streams The two-stream network consists of two VGG16 models trained for face detection, concatenated with a FC layer at the end that reduces the dimensionality, before the two neurons that output the pitch and yaw prediction. There are two versions of two streams:

1. Eyes version This is the version used by the original experiments in [22], where the inputs of the network are:

- The face of the person for which the gaze vector is to be estimated.
- The cropped eyes of the person in this frame.
- The face landmarks of the person in this frame.

The landmarks are an optional input feature to the model and are concatenated with the output of each stream as the input of the first fully connected layer of 5836 neurons. This model is used by the next experiments:

- Normalized face, eyes, landmarks with 5836 neurons in the las fully connected layer (referred to as NFEL5836).

- Normalized face, eyes, landmarks with 2918 neurons in the las fully connected layer (referred to as NFEL5836_2918).
- Recurrent model of NFEL5836_2918 with a GRU as a tepomporal layer (referred to as NFEL5836GRU).

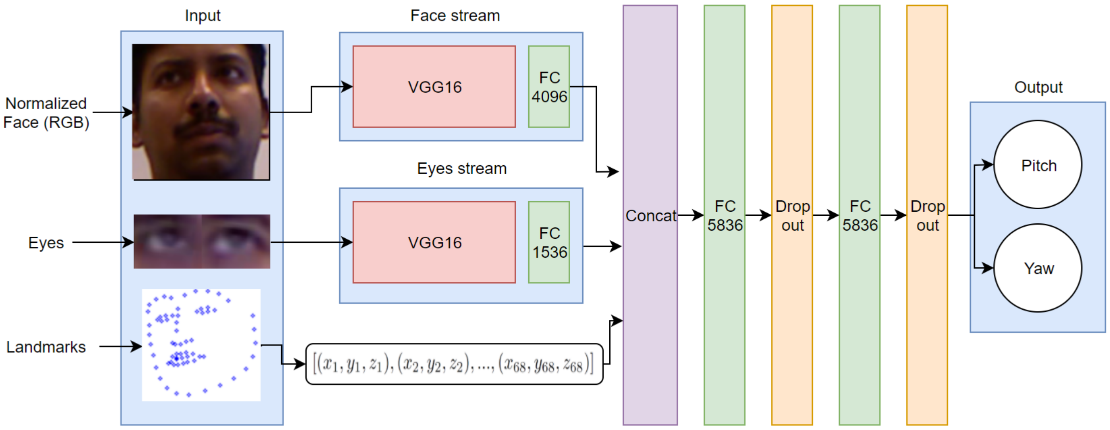


Figure 12: Architecture of the two stream network model for face, eyes and optionally landmarks

2. Optical flow version This is the most basic model that adds optical flow as an input feature, and we use it to obtain a baseline of the performance of the experiments with optical flow. The experiments realized with this model have as input the normalized face and the optical flow of the face of the current frame with the previous one, also normalized.

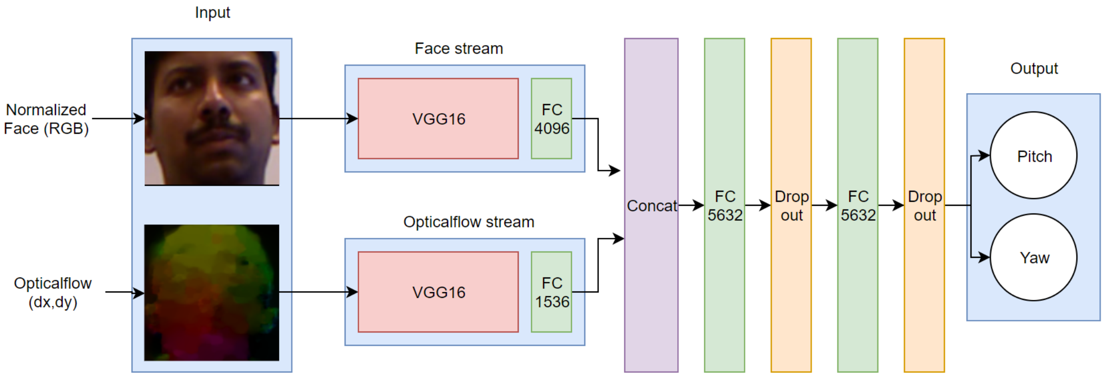


Figure 13: Architecture of the two stream network model for face and optical flow.

This model is used by the NFO5632 experiment in Section 5.7.1.

Three streams This model adds the eyes as input feature to the two-stream model with the intention to recover this lost feature.

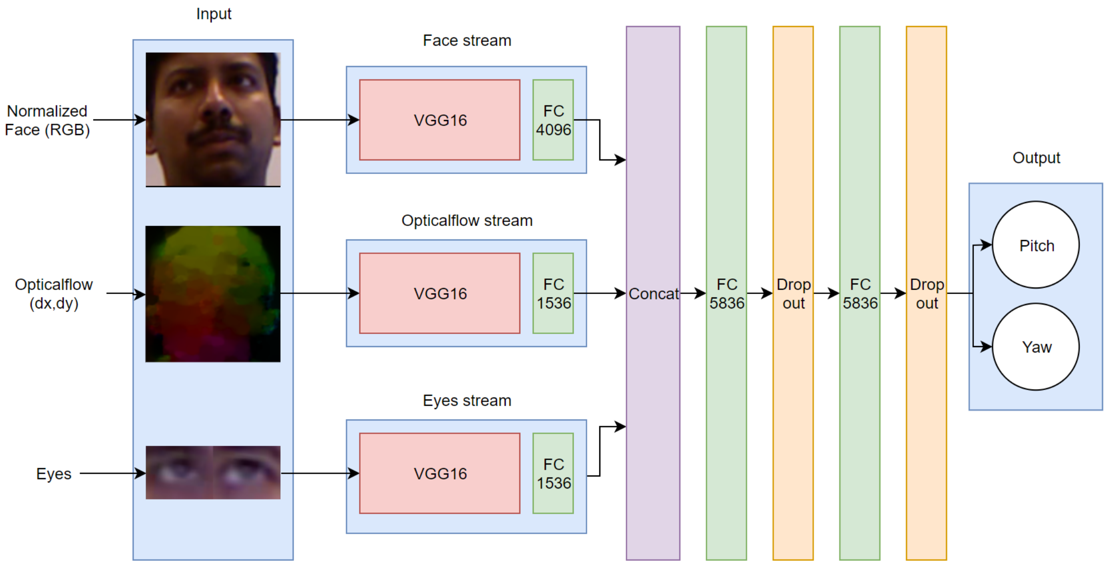


Figure 14: Architecture of the three stream network model for face, eyes and optical flow.

This model is used by the NFEOF5632 experiment in Section 5.7.2.

Three streams landmarks This model also adds the landmarks as an input feature to the three-stream model with the intention to add a geometric contraint to the networks.

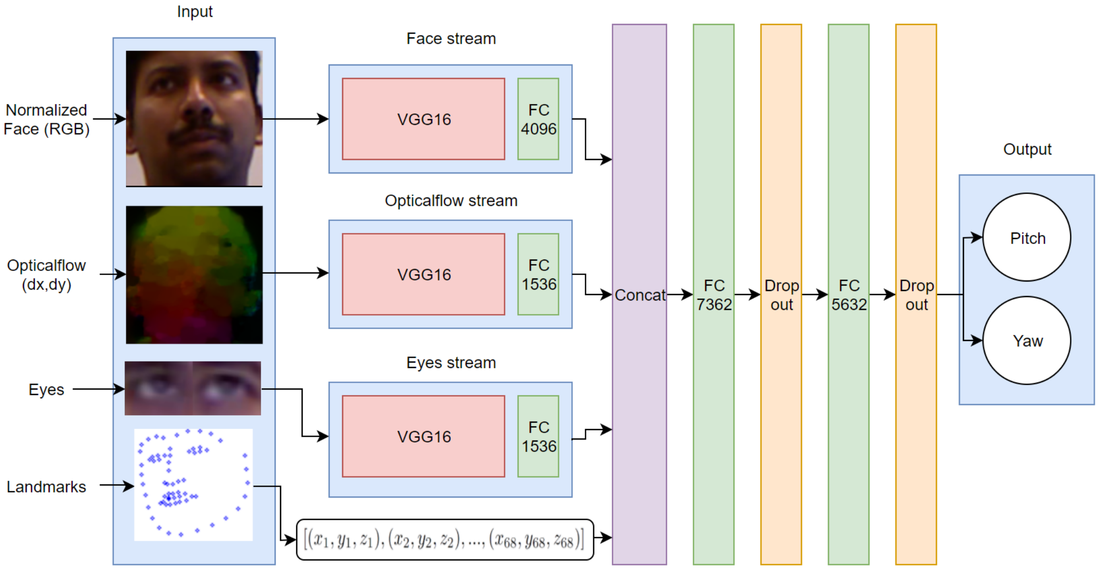


Figure 15: Architecture of the three stream network model for face, eyes, optical flow and face landmarks.

This model is used by the NFEOF5632 experiment in Section 5.7.2.

Four streams This is the most complete and complex model with all the possible features extracted (Normalized face, optical flow from the face, eyes, optical flow from the eyes, and face landmarks).

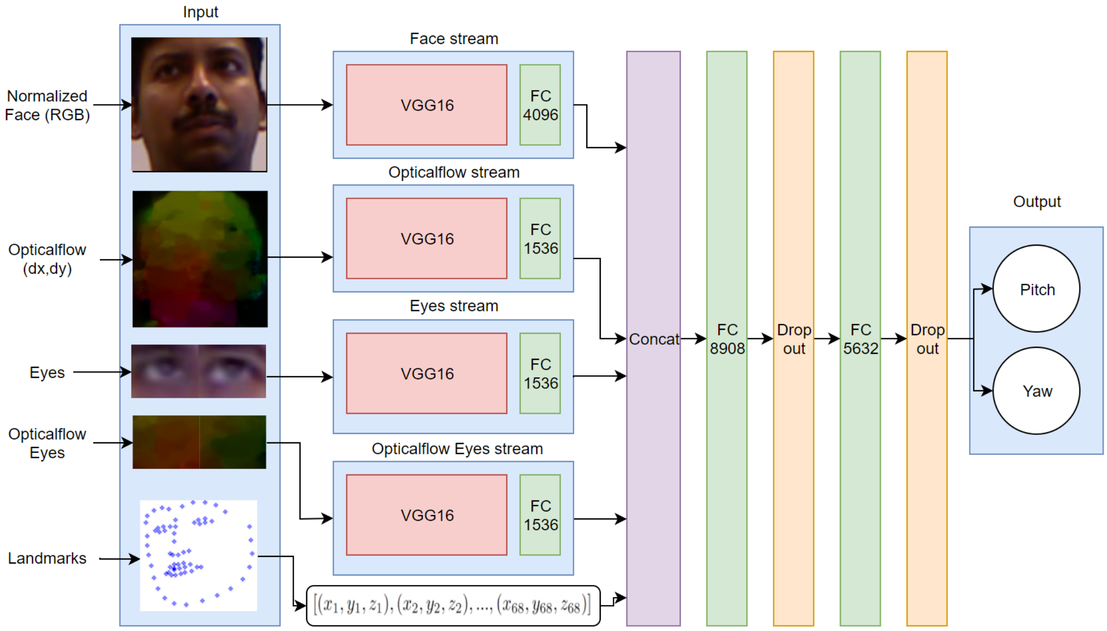


Figure 16: Architecture of the four stream network model for face, eyes, optical flow of face, optical flow of eyes and face landmarks.

4.4.2 Recurrent model

In this section, we show the recurrent model that we are going to use in the experiments. This model has as input the sequential information resulting from the feature vectors of each frame. It is a many-to-one recurrent network.

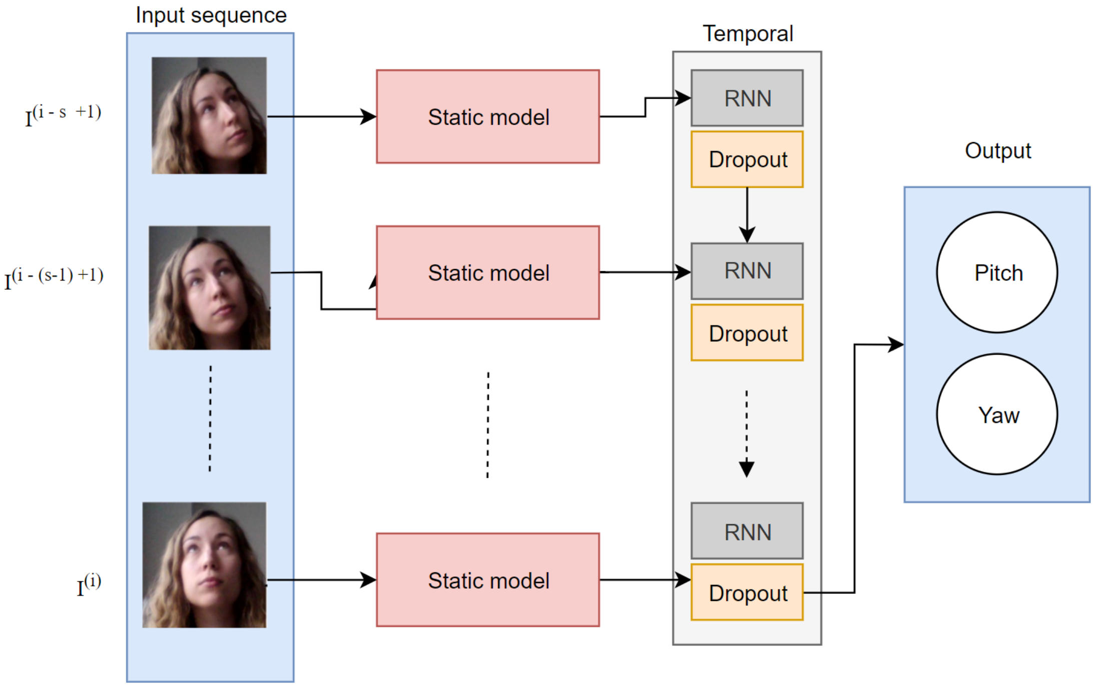


Figure 17: Recurrent model used in the temporal experiments. Face images extracted from [22].

During the training, the static model is frozen except for the last fully connected layers. Allowing learning in the last layers helps to adapt the output of the static models for a better representation that helps the recurrent layer.

The static model could be any of the previous ones as long as the input is adapted to the model.

5 Experiments

Our main goal is to evaluate whether the addition of optical flow improves gaze estimation accuracy in comparison to other static and spatio-temporal approaches. The metric we used to measure how well a model performed is the angular error. This measures the average angular error between the vector predicted and the ground truth. In the equation 4 could be seen how is computed.

$$\text{angular error } = arccos(\frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

Where **A** could be the estimated vector and **B** the ground truth vector.

5.1 Hardware

To do these experiments, we used two computers:

1. AMD Ryzen 3900X with 24 execution threads, 32GB of RAM, 2TB of SSD and 2TB of HDD, 1x NVIDIA TITAN (Maxwell) X 12GB VRAM.
2. AMD Ryzen 2600X with 12 execution threads, 64GB of RAM, 4TB of HDD, 2x NVIDIA G-FORCE 1080ti 11GB VRAM.

These computers took a total of 1.5 months to compute all the experiments working in parallel.

5.2 Data

The final size of the dataset after processing all the data was 1.5TB of disk space. Specifically, it is distributed as follows:

- The EYEDIAP dataset with the subjects of the floating target part occupied 300GB of disk space in 640x480 RGB BMP images. We had to compress them in PNG, in which they only needed 70GB. Converting from PNG over BMP also helps with the loading time from an HDD. The decompression of a PNG file is faster than the reading of the equivalent image in BMP format from an HDD.
- The optical flow vectors of the 66 sessions of the EYEDIAP dataset took 1,2TB of disk space because the matrices storing the displacements of the pixels in x and y should be float32 or the small movements of pixels will not be represented with enough precision. We also make a cache with the trimmed part of the face (250x250x2 floats) and the eyes part (120x48x2 floats) that gave us a 10x speed increase and a space reduction remaining in only 80GB of disk space.

Experiment names We had to name the experiments to make the plots more clear. We used the following pattern <letters meaning the inputs of the network><number of neurons in the last fully connected layer> [optional comment relevant for the experiment]. The possible meanings of the letters are the next: N - Normalized, F - Face, E - Eyes, O - Opticalflow, L - Landmarks of the face.

For instance:

- NF4096, this means Normalized Face as input and 4096 neurons in the last fully connected layer.
- NFEL5632, this means Normalized Face, Eyes, landmarks, and 5632 neurons in the last fully connected layer.
- NFEL5632GRU, the comment at the end of the name in this experiment means it is a recurrent one, and for the still part of the model is using the weights of NFEL5632.

Folds We decided to do four-fold cross-validation because four divisions are the best we can do with the 16 subjects for the floating target of the EYEDIAP dataset. In particular, we used the following splits:

- **Fold 1**: 5_A_FT_S, 5_A_FT_M, 10_A_FT_S, 10_A_FT_M, 11_A_FT_S, 11_A_FT_M, 14_A_FT_S, 14_A_FT_M, 14_B_FT_S, 14_B_FT_M.
- **Fold 2**: 1_A_FT_S, 1_A_FT_M, 4_A_FT_S, 4_A_FT_M, 6_A_FT_S, 6_A_FT_M, 15_A_FT_S, 15_A_FT_M, 15_B_FT_S, 15_B_FT_M.
- **Fold 3**: 2_A_FT_S, 2_A_FT_M, 3_A_FT_S, 3_A_FT_M, 8_A_FT_S, 8_A_FT_M, 16_A_FT_S, 16_A_FT_M, 16_B_FT_S, 16_B_FT_M.
- **Fold 4**: 7_A_FT_S, 7_A_FT_M, 9_A_FT_S, 9_A_FT_M, 12_B_FT_S, 12_B_FT_M, 13_B_FT_S, 13_B_FT_M.

5.3 Input pre-processing

During training, the original image is pre-processed to get the two normalized input images. The normalized whole-face patch is centered 0.1 meters ahead of the head center in the head coordinate system and is defined such that the image has a size of 250×250 pixels. The difference between this size and the final input size allows us to perform random cropping and to zoom to augment the data. Similarly, each normalized eye patch is centered in their respective eye center locations. In this case, the virtual camera matrix is defined so that the image is cropped to 70×58, while in practice, the final patches have a size of 60×48.

The augmentation step consist in:

- Apply horizontal flip in random frames.
- Apply a shift displacement to the image in x and y.
- apply a zoom in range [0.98 to 1.02].
- Modify the brightness conditions.
- Add a Gaussian noise of a 3

The optical flow is pre-processed with the same method that is used in the original face, so each pixel containing (dx, dy) displacements corresponds with the correct pixel in the normalized image.

Some of the frames do not have a valid ground truth vector. We discarded the frames with some of the following problems:

- Face landmarks not detected

- Subject not looking at the target
- 3D head pose, eyes or target location not properly recovered
- Eyeballs rotation violating physical constraints ($|\theta| \leq 40°, |\phi| \leq 30°$).

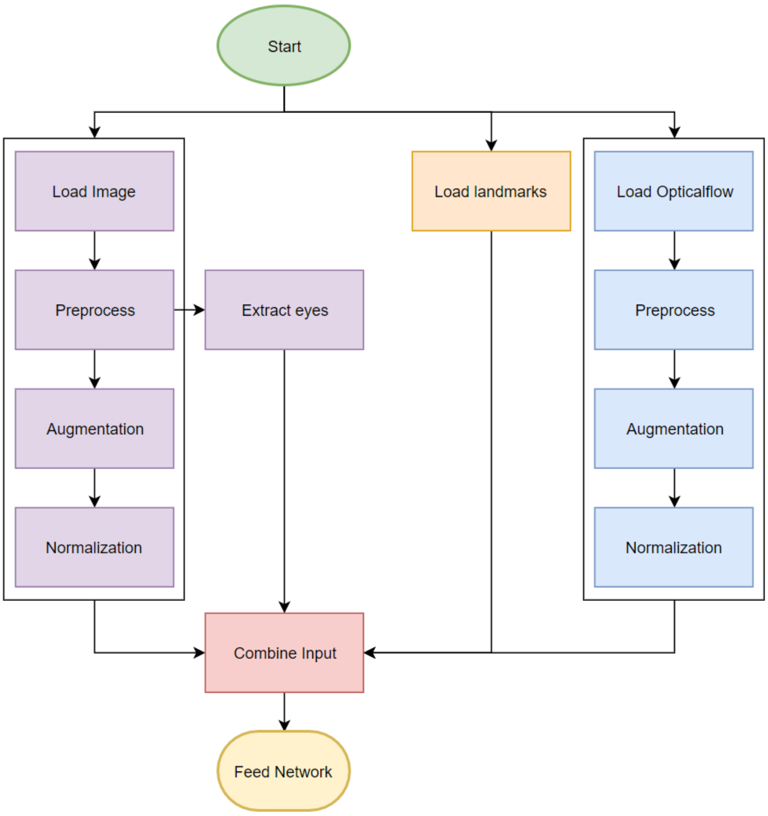


Figure 18: Diagram of the flow of the data generation.

5.4 Hyperparameters

We fixed the next hyperparameters for all the experiments:

Learning rate In our experiments, we tried three learning rates, **0.001**, **0.0001**, and 0.00001. These learnings rates were empirically pre-selected, observing the descending of the angular error in the first epochs of the experiments. The experimentation shows that **0.0001** is a good learning rate for all the experiments. Fine-tuning of this parameter could be interesting to do a best final and more accurate model.

Optimizer The optimizer in all the experiments was Adam [14]. Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments, is computationally efficient, has little memory requirements, and works in problems that are large in terms of data and parameters. The main advantage of Adam it computes an adaptive learning rates for each parameter that is tuned.

Batch size For the batch size, we tried with 8, 16, 24, 32, 48, 64, but not all the experiments could be trained using the bigger ones because the memory of the GPU was not enough to copy the train data in the 12GB of VRAM. In the appendix, "Complete history logs of the experiments." the loss and angular error tried for all the experiments could be seen. For the sake of the comparative, we decided to go with a batch size of 8 for mainly two reasons: (i) the temporal experiments only work with this batch size, and (ii) The best angular error was obtained with this batch size. Obtain a better accuracy with a smaller batch size is the effect of a less averaged vector with a more significant norm. It also makes the training noisier, allowing the loss to arrive at a better local minimum. Instead, with a bigger batch size the averaged gradient would result in smaller steps. In [1] this effect is analyzed.

Dropout We empirically set a dropout with a probability of **0.3** in the last layers of the models. Dropout is a regularization technique that approximates training a neural network simulating different architectures in parallel. Its usually placed between two fully connected layers with the objective of reducing overfitting. Works in the following way: during training, some connections between two layers are erased with a probability p.

5.5 Software

The software used to code the experiments was:

- Python=3.5.6
- tensorflow=1.10
- keras=2.1.5
- keras_vggface=0.5
- numpy=1.14.2
- opencv=3.4.0.14

The base models are the ones provided in the Keras VGGFace package[5]. This package includes several famous deep neural networks pre-trained to do face recognition.

5.6 Baseline results

The first thing that we should check is if we can reproduce the results of the experiments of [22]. To do that, we recreated the exact same environment very carefully using the same Python and packages versions.

5.6.1 Normalized face, 4096 neurons

This is the first experiment that we test, and the most basic one, the input of the network, consists only in the face normalized. The configuration of this experiment is:

- Model: One stream, see Section 4.4.1.
- Fully connected layers neurons: 4096
- Input: Only generates faces

For this experiment the best result are in the next table:

experiment	fold	batch_size	val_angle_error	epoch
NF4096	1	8	5.02703	16
NF4096	2	8	5.14182	12
NF4096	3	64	4.90789	20
NF4096	4	8	6.68591	11

Table 1: Results of the best epochs of the experiment NF4096 for the four folds.

5.6.2 Normalized face, eyes and landmarks, 5836 neurons

This experiment is the most complete of the experiments without optical flow and aims to test if using all the features extracted improves the basic model given better results. The input of the network consists of the face normalized, the eyes extracted, and landmarks.

The configuration of this experiment is:

- Model: Two stream (with landmarks) 4.4.1
- Fully connected layers neurons: 5836
- Input: Generates face, eyes and landmarks.

For this experiment the best result are in the next table:

experiment	fold	batch_size	val_angle_error	epoch
NFEL5836	1	8	5.09876	11
NFEL5836	2	8	5.14995	21
NFEL5836	3	8	4.99211	21
NFEL5836	4	64	6.92378	21

Table 2: Table of the best epochs of the experiment NFEL5836.

5.6.3 Normalized face, eyes, and landmarks, 2918 neurons

This experiment is the same as the previous one 5.6.2, only changing the last fully connected layer to 2918 neurons (exactly the number of neurons of the previous fully connected divided by two). The reasons to do this experiment are the following ones:

- To have a smaller output, so the recurrent models that use this model as the static part could be trained more easily since the recurring models explored have been very slow to train (3x times a static model).
- Explore the impact of the reduction of the last fully connected layer before the two neurons that makes the output prediction.

For this experiment the best result are in the next table:

experiment	fold	batch_size	val_angle_error	epoch
NFEL5836_2918	1	8	5.11546	12
NFEL5836_2918	2	8	5.23134	15
NFEL5836_2918	3	8	4.81456	20
NFEL5836_2918	4	32	6.36455	12

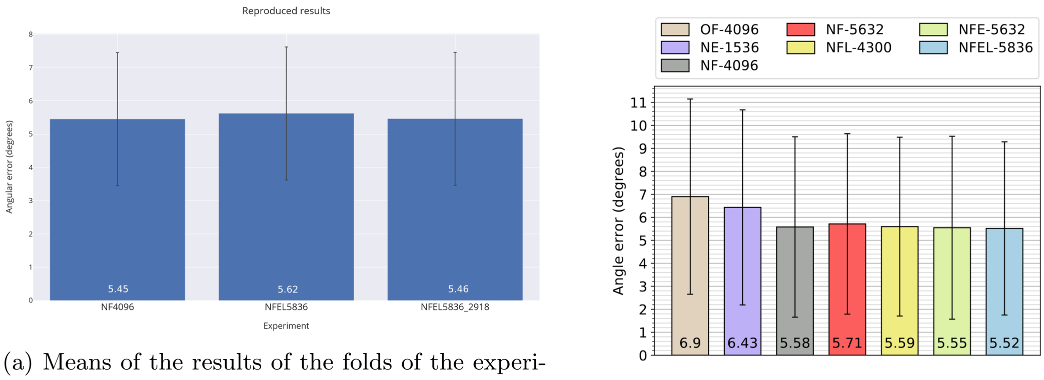
Table 3: Table of the best epochs of the experiment NFEL5836_2918

5.6.4 Baseline summary results

Finally, we can compare the results that we obtained. We can observe in Table 4 that the experiments with more features performs slightly better but not to say that they are clearly better.

Fold	NFEL5836_2918	NFEL5836	NF4096
1	5.11546	5.09876	**5.02703**
2	5.23134	5.14995	**5.14182**
3	**4.81456**	4.99211	4.90789
4	**6.36455**	6.92378	6.68591

Table 4: Results of the four fold of all the experiments together, in bold the lower error of each one.



(a) Means of the results of the folds of the experiments that we reproduced.

(b) Final results of [22].

Figure 19: Plots of the means with the standard deviation obtained in all the experiments.

We can see how selecting a batch size of 8 give us better accuracy in the simplest experiment (NF4096). These errors are what we are going to take as a baseline to see if optical flow improves the results.

5.7 Optical flow experiments

In this section, we show the experiments that we design to see how the addition of the optical flow information affects in the angular error of the experiments using the models previously defined. All the optical flow experiments have the last fully connected layer before prediction fixed to 5632 neurons since we saw in the previous section that the experiment with 2918 neurons performed as good as the others and did not impact the accuracy of the experiments. However, the first fully connected layer changes its size depending on the inputs.

5.7.1 Normalized face, Optical flow, 5632 neurons

This experiment is very similar to the first experiment that we test in Section 5.6.1, the main modifications are, changing the model for the two-stream one defined in the Section 4.4.1, and increasing the number of neurons in the last fully connected layer before the output. The input of the network consists of the face normalized and the optical flow of that normalized face computed using the previous frame in time. The configuration of this experiment is:

- Model: Two stream for optical flow 4.4.1
- Fist fully connected layer neurons: 5632
- Last fully connected layers neurons: 5632
- Input: Generates faces and optical flow of the face.

For this experiment the best result are in the next table:

experiment	fold	batch_size	val_angle_error	epoch
NFO5632	1	8	4.74253	6
NFO5632	1	48	4.84704	16
NFO5632	2	8	5.34078	4
NFO5632	2	32	5.24506	20
NFO5632	3	8	4.55839	16
NFO5632	4	8	6.65459	4

Table 5: Table of the best epochs of the experiment NFO5632

5.7.2 Normalized face, eyes, Optical flow, 5632 neurons

In this experiment, we decided to add the information of the eyes to the previous one. To do this, we had to make a new model the three-stream optical flow that can be seen in Section 4.4.1.

- Model: Three stream for optical flow 4.4.1.
- Fist fully connected layer neurons: 7168.
- Last Fully connected layers neurons: 5632.
- Input: Generates faces, eyes, and optical flow of the face.

For this experiment the best result are in the next table:

experiment	fold	batch_size	val_angle_error	epoch
NFEOF5632	1	8	4.50474	20
NFEOF5632	2	8	5.49019	12
NFEOF5632	3	8	4.70594	18
NFEOF5632	4	8	6.60661	11

Table 6: Table of the best epochs of the experiment NFEOF5632

5.7.3 Normalized face, eyes, landmarks and Optical flow, 5632 neurons

In this experiment, we decided to add the information of the landmarks to the previous one. To do this, we had to modify the three-stream optical flow model also to accept the landmarks information that can be seen in Section 4.4.1.

- Model: Three streams for optical flow with landmarks 4.4.1
- Fist fully connected layer neurons: 7362
- Last Fully connected layers neurons: 5632
- Input: Generates faces, eyes, landmarks, and optical flow of the face.

For this experiment the best result are in the next table:

experiment	fold	batch_size	val_angle_error	epoch
NFELOF5632	1	8	4.71119	18
NFELOF5632	2	8	5.67651	20
NFELOF5632	3	8	4.63749	12
NFELOF5632	4	24	6.55591	15

Table 7: Table of the best epochs of the experiment NFELOF5632

5.7.4 Normalized face, Optical flow, Resnet, 5632 neurons

In this experiment, we decided to change the VGG network of the models by a resnet50, hoping that the more expression power of the network helps to achieve better results. The base experiment is the NFO5632 since it is not clear that adding the eyes and landmarks to the input are making better predictions.

- Model: Two streams for optical flow 4.4.1, but instead of the VGG network for the two streams, we use resnet50, also pre-trained for face detection.
- Fist fully connected layer neurons: 5632
- Last fully connected layers neurons: 5632
- Input: Generates faces and optical flow of the face.

For this experiment the best result are in the next table:

experiment	fold	batch_size	val_angle_error	epoch
NFO5632RESNET	1	8	4.25655	5
NFO5632RESNET	2	8	5.77746	12
NFO5632RESNET	3	8	4.34516	18
NFO5632RESNET	4	8	6.53762	12

Table 8: Table of the best epochs of the experiment NFO5632RESNET

5.7.5 Normalized face, Optical flow, Resnet, 2918 neurons

Modified the previous experiment to have 2918 neurons in the last fully connected layer. Preparing this model to be the static part of the recurrent one.

- Model: Two streams for optical flow 4.4.1, but instead of the VGG network for the two streams, we use resnet50, also pre-trained for face detection.
- Fist fully connected layer neurons: 5632
- Last fully connected layers neurons: 2918
- Input: Generates faces and optical flow of the face.

For this experiment the best result are in the next table:

experiment	fold	batch_size	val_angle_error	epoch
NFO5632RESNET_2918	1	16	4.72577	18
NFO5632RESNET_2918	2	16	5.71042	10
NFO5632RESNET_2918	3	16	4.46826	13
NFO5632RESNET_2918	4	16	5.99694	8

Table 9: Table of the best epochs of the experiment NFO5632RESNET_2918

5.7.6 Comparing results

To be able to compare the results of the experiments with the optical flow, we followed two methodologies.

Fold	NFO5632	NFEOF5632	NFELOF5632	NFO5632RESNET	NFO5632RESNET_2918
1	4.74253	4.50474	4.71119	**4.25655**	4.72577
2	**5.24506**	5.49019	5.67651	5.77746	5.71042
3	4.55839	4.70594	4.63749	**4.34516**	4.46826
4	6.65459	6.60661	6.55591	6.53762	**5.99694**

Table 10: Table with the results of each fold for each experiment.

Do the mean of all the frames for each experiment.

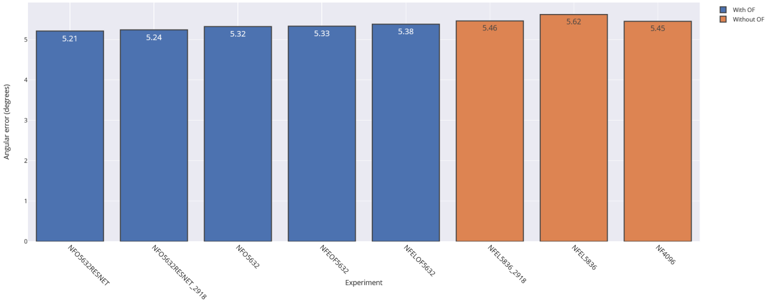


Figure 20: Mean angular error of all the frames in the dataset for each experiment.

Experiment	Mean	Standard deviation
NFO5632RESNET	5.21	3.74
NFO5632RESNET_2918	5.24	3.61
NFO5632	5.32	3.67
NFEOF5632	5.33	3.92
NFELOF5632	5.38	3.81
NFEL5836_2918	5.46	3.86
NFEL5836	5.62	3.92
NF4096	5.45	3.86

Table 11: Mean and standard deviation of the experiments with the estimations of all the frames of the dataset.

We can observe that all the experiments with opticalflow perform better, and the simplest model (normalized face and optical flow of the face) are the ones that obtain better accuracies. And comparing the base models the experiments that use ResNet50 also perform better. We can conlcude that opticalflow improved the performance in a range of 1.3% to 7.3%.

5.8 Recurrent experiments

In this section, we explore the effects of adding temporal information to the already trained models for gaze estimation from still images. To do this, we use a new recurrent model explained in Section 4.4.2. First, we are going to reproduce the results of [22], and then we analyze the effects of converting the models with the optical flow (that they already have temporal information) in recurrent ones. All the recurrent models are trained using the same four-fold cross-validation explained in the Section 5.2. The weights of the static models are always the weights of the corresponding fold.

5.8.1 Baseline experiment

As in the models of the still image, we want first to reproduce the results of [22] to ensure that we are using the same metrics to compare the experiments. In this case, there is only one experiment.

Normalized face, eyes, landmarks, recurrent with GRU layer. We are reproducing the results of this experiment. Each fold has been trained using the best fold of the still image NFEL5836_2918 model (see, Section 5.6.3).

- Model: Recurrent model 4.4.2, Frozen pretained model NFEL5836_2918 5.6.3.
- Recurrent layers: 1 GRU with 128 units.
- Input: Generates from 4 consecutive frames, faces, eyes and landmarks.

This experiment was trained using the four folds defined in Section 5.2 and the mean using only the folds with the minimum error was **5.29** while each fold individually performed in the following way:

experiment	fold	batch_size	val_angle_error	epoch
NFEL5836GRU	1	8	4.57239	6
NFEL5836GRU	2	8	5.43925	20
NFEL5836GRU	3	16	4.59519	8
NFEL5836GRU	4	8	6.57170	1

Table 12: Table of the best epochs of the experiment NFEL5836GRU

5.8.2 Optical flow recurrent experiments

In this section, we explore the effects of the use of a recurrent neural network. Like in the previous section, we use the model defined in Section 4.4.2. We are using a network already trained of one of the best models of optical flow for the still image gaze estimation part of the model. The objective of this experiment is to see if they improve the angular error when having both the temporal information provided by the optical flows of the frames and the temporal information that provides the sequence of frames by itself.

Normalized face, optical flow face, recurrent with GRU layer. This experiment uses one of the best optical flow models for the still image gaze estimation for the frozen part of the recurrent model.

This experiment was trained using the four folds defined in the Section 5.2 and the mean using only the folds with minor error was **5.28** while each fold individually performed in the following way:

experiment	fold	batch_size	val_angle_error	epoch
NFO5632GRU	1	8	4.55079	6
NFO5632GRU	2	8	5.15773	13
NFO5632GRU	3	8	4.61644	2
NFO5632GRU	4	8	6.79343	6

Table 13: Table of the best epochs of the experiment NFO5632GRU

5.8.3 Summary results

The recurrent experiments gave an angular error of **5.29** in the case of the baseline experiment NFEL5836GRU and **5.28** in the case of NFO5632GRU experiment.

These results compete with the ones obtained in the static experiments (NFO5632RESNET and NFO5632RESNET_2918) that uses as temporal information the optical flow of the face. However, this does not mean that recurrent models are not working or can achieve better results. The only valid conclusion that we can extract is with these hyperparameters (Units in the recurrent layer, type of the recurrent layer, learning rate, batch size, optimizer); the experiments do not give better accuracy than the static ones with optical flow and a resnet50 base model.

5.9 Filter experiments

In this chapter, we analyze the effects of adding a median filter to the model estimation. First, we have to define what is the median vector. We used the following definition.

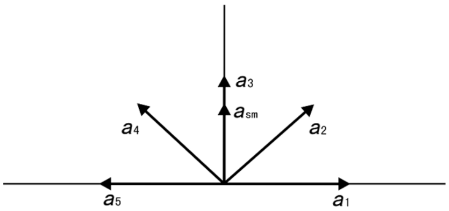


Figure 21: Image extracted from [16]. Comparison of the vector median vector and the scalar median vector. The median vector a_3 is a member of the input set of vectors, whereas the scalar median vector a_{sm} found by taking the scalar median of each component of vectors in the input set is not a member of the input set.

5.9.1 Median vector

The standard definition of a median value is: Given a set of scalars $S_i = \{a_i\}\, i = 1, 2, \ldots, N$, if the set is sorted, then the middle value of the sorted set is the median. This sorting-based definition is intuitive and easy to understand, but hard to extend to a set of vectors. In the work of Liu *et al.* [16], they redefine the scalar median value based on a minimum-distance concept. The median member a_m, according to the minimum-distance definition, is the member whose distance to all other members in the set is smallest. This definition can be expressed as:

$$a_m = \underset{a_m \in S_i}{\operatorname{argmin}} \sum_{i=1}^{N} \|a_m - a_i\|_L \tag{5}$$

Where i is the summation index, N is the number of members in the set, and L denotes the order of the norm. Any proper norm (e.g., L_1, L_2 or L_∞) is eligible to be used in this definition. In our particular case, we choose L_2.

5.9.2 Window Filter

The filter we apply consist in instead of predicting exactly the last frame recorded from the camera, we compute the estimation for each frame in a window of frames. Once we have the estimated vectors per frame, we apply a medium window size filter w, and taking into account the values of this window we take the median vector, this being the value of the central frame of the window.

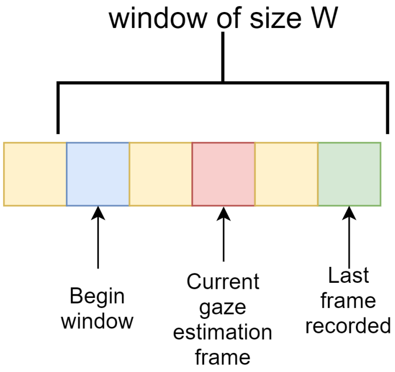


Figure 22: Illustrating the sliding window that we use in the filter.

5.9.3 Experiments

We decided to experiment using one of the best models we obtain during the experiments of gaze estimation with still images. In particular, we choose to do the test with the NFO5632RESNET model, see Section 5.7.4. The evaluation methodology is similar to other experiments. We use the validation data of each fold to compute the predictions and then over these predictions apply the window filter before computing the angular error with the ground truth.

Fold	Window size			
	Without filter	3	5	7
1	4.246263	4.2088714	4.189559	4.190738
2	5.624934	5.597168	5.578781	5.586626
3	4.3602552	4.327826	4.3174233	4.3054256
4	6.570273	6.5285983	6.516427	6.5036163
mean	5.2	5.17	5.15	5.15

Table 14: Validation angular error when a median filter is applied to the experiment NFO5632RESNET in the four folds.

It can be observed that in all cases applying a median filter with a window of 3 to 7 improves the accuracy of the estimate by 1%.

5.10 Eye Ablation studies

In this section, we show how by eliminating the eyes from the cutout of the face, the network is still able to estimate the gaze, although with a larger error. We have to remember that the EYEDIAP dataset has the moving head part, which is composed of subjects who are not looking towards the target is located. See Section 3.

The ablation of the eyes was made using the already computed face landmarks replacing with two black rectangles, the eyes regions.

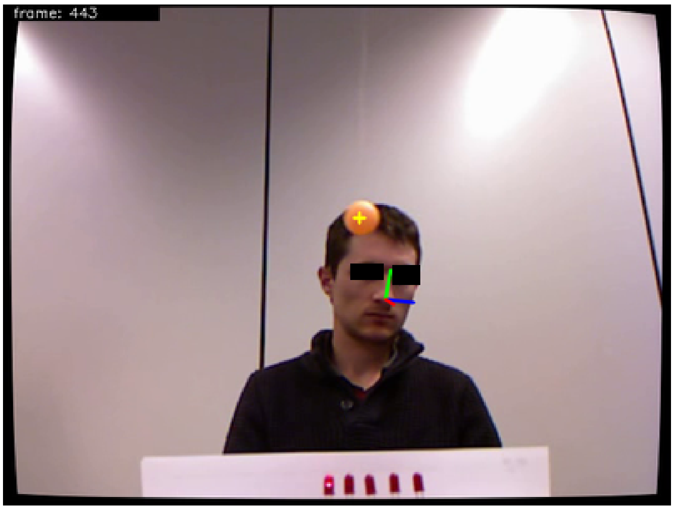
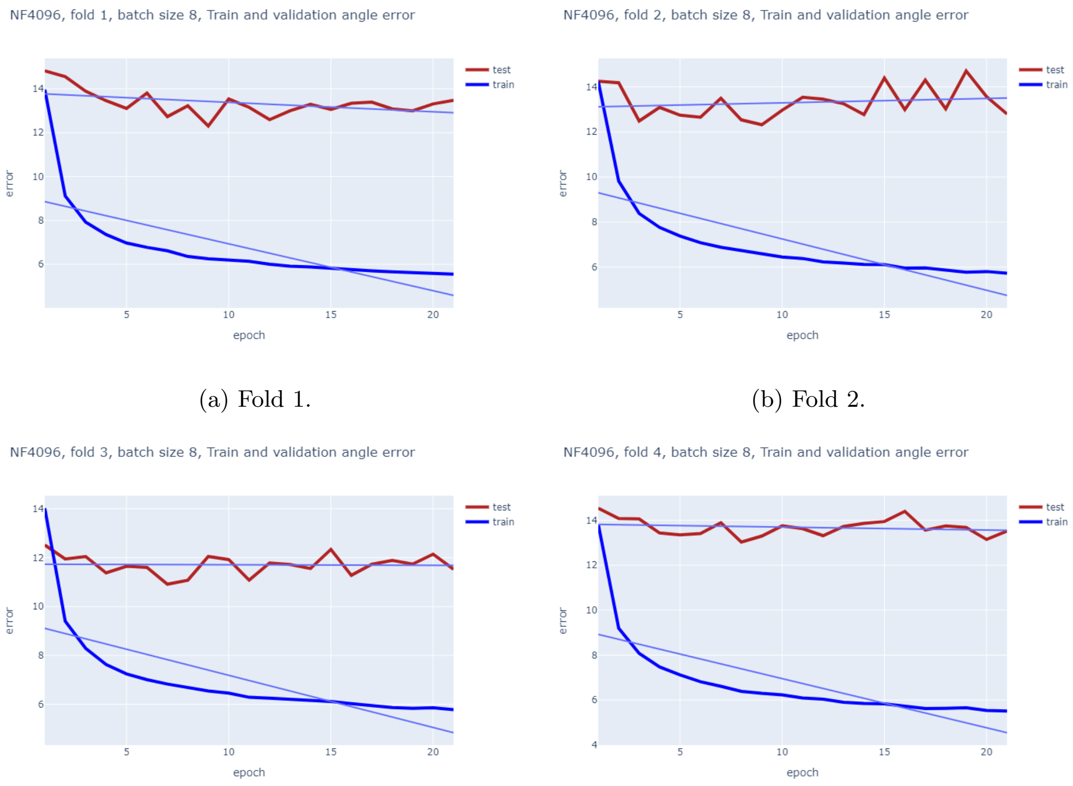


Figure 23: Subject 1_A_FT_M of the EYEDIAP dataset with eyes ablation.

A new experiment like the NF4096 was trained with a dataset where the eyes were ablated, and the experiments show that the accuracy is still relevant and is as good as some of the techniques reviewed in the Related work Section 2.



(a) Fold 1.

(b) Fold 2.

(c) Fold 3.

(d) Fold 4.

Figure 24: Plots of the train and test angle error for the experiment NF4096 using the eyes ablation dataset.

experiment	fold	batch_size	val_angle_error	epoch
NF4096	1	8	12.30331	9
NF4096	2	8	12.31694	9
NF4096	3	8	10.91418	7
NF4096	4	8	13.02588	8

Table 15: Table of the best epochs of the experiment NF4096 with ablation of the eyes.

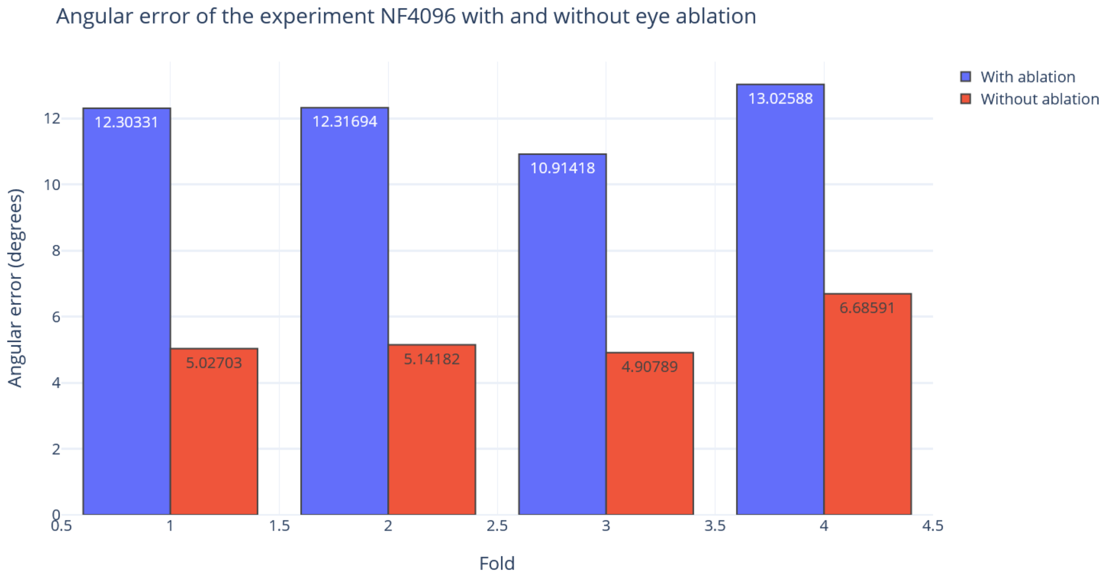


Figure 25: Comparison of the angular error for the experiment NF4096 between the original dataset and the dataset with eye ablation.

6 Ethics

Since there cannot be a future responsible for artificial intelligence without ethics, in this section we analyze the moral implications of this work, although without going into much depth. The ethical considerations do not come from the new model or the comparative we make, but from the uses that can be given to the models developed in this master thesis and the previous or related works.

For the training data, we are using a dataset of people that gave their consent to be recorded and use this information to do academic research. So in principle, there is no problem with training a model that is based on their data. We have to remember that we are using the whole face to predict the gaze and not only the eyes. The personal information of the subjects will be codified in the models.

This technology could be applied without us knowing it, and this could produce undesired uses without our consent. For example private companies can do gaze analysis to determine people interests and adapt establishments to obtain economic revenues. Collecting information about what products you watch more, what gets your attention, so they can optimize the organization of their stores and research what things that get people's attention to sell more. But they can also sell your private information to other companies for worse purposes.

Of course, immoral uses of this technology will occur as well. The desire that companies have to collect data nowadays will make them try to collect more data. For instance, how people interact with their phones, analyze how two persons are interacting and infer the type of conversation they are having (without taking into account other techniques like natural language processing and extracting information from the video), and this could be used to make more addictive games or applications.

In the other hand, there is much room to do of the world a better place, just for good. For instance, applications that can help in the wellness of the people, if you have a mobile phone equipped with this technology you could make applications that can detect if you are having a sad moment while using your mobile phone chat applications. The app can give you a tip to make you feel better or if it detects suicidal thoughts, prevent them. It can also be useful for helping people to improve their communication skills thanks that with a simple video, they can analyze human-human interactions.

In conclusion, we can only hope that the Governs legislate about the advances made with artificial intelligence so that they are used mainly for beneficial purposes for the world and not misused.

7 Conclusions and future work

In this work, we studied the addition of optical flow at the combination of full-face and eye images along with facial landmarks for person and head pose-independent 3D gaze estimation. We proposed several multi-stream CNN networks and one recurrent model that utilizes sequential eye and head movement with optical flow information. After that, we analyze the effects of applying a median filter to the output vector using several window sizes. Besides, an eye ablation study has also been carried out to analyze how much information the models could extract only from the appearance of the face.

We showed that adding optical flow to the baseline experiments outperform the accuracy in almost all input combinations, and the use of the resnet50 as a base model for the streams was a better choice than the VGG16. We also showed that applying a median filter could improve the accuracy of the models that already have temporal information of optical flow.

To the best of our knowledge, this is the first attempt to use optical flow as temporal information in the context of gaze estimation from remote cameras. As future work, a better fine tuning of the hyperparameters of the models could be done, also a recurrent model using ConvLSTM instead of GRU could be interesting, and try 3DCNNs as well to encode the deep features.

References

[1] Effect of batch size on training dynamics. `https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e`. Accessed: 2020-01-19.

[2] Shumeet Baluja and Dean Pomerleau. Non-intrusive gaze tracking using artificial neural networks. In *Advances in Neural Information Processing Systems*, pages 753–760, 1994.

[3] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, Feb 2005.

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

[5] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

[6] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition, 2016.

[7] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[8] Kenneth Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: a database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. pages 255–258, 03 2014.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Quentin Guillon, Nouchine Hadjikhani, Sophie Baduel, and Bernadette Rogé. Visual social attention in autism spectrum disorder: insights from eye tracking studies. *Neuroscience and Biobehavioral Reviews*, 42, 05 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[12] Hardik Jain. *Using Morphable Face Model to Improve Stereo Reconstruction and Visualising the Model on a Smartphone*. PhD thesis, 05 2016.

[13] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild, 2019.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[15] Ce Liu et al. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[16] Yike Liu. Noise reduction by vector median filtering. *GEOPHYSICS*, 78(3):V79–V87, 2013.

[17] Simon Liversedge and John Findlay. Saccadic eye movements and cognition. *Trends in Cognitive Sciences*, 4:6–14, 02 2000.

[18] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. In *2011 International Conference on Computer Vision*, pages 153–160, Nov 2011.

[19] C. H. Morimoto, A. Amir, and M. Flickner. Detecting eye position and gaze from a single camera and 2 light sources. In *Object recognition supported by user interaction for service robots*, volume 4, pages 314–317 vol.4, Aug 2002.

[20] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation, 2016.

[21] Yumiko Otsuka, Andrew Calder, and Colin Clifford. Dual-route model of the effect of head orientation on perceived gaze direction. *Journal of experimental psychology. Human perception and performance*, 40, 04 2014.

[22] Cristina Palmero, Javier Selva, Mohammad Ali Bagheri, and Sergio Escalera. Recurrent CNN for 3d gaze estimation using appearance and shape cues. *CoRR*, abs/1805.03064, 2018.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[24] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1828, June 2014.

[25] Kar-Han Tan, David J Kriegman, and Narendra Ahuja. Appearance-based eye gaze estimation. In *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.*, pages 191–195. IEEE, 2002.

[26] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2014.

[27] Wang, Sung, and Ronda Venkateswarlu. Eye gaze estimation from a single image of one eye. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 136–143 vol.1, Oct 2003.

[28] K. Wang and Q. Ji. Real time eye gaze tracking with 3d deformable eye-face model. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1003–1011, Oct 2017.

[29] Kang Wang, Hui Su, and Qiang Ji. Neuro-inspired eye tracking with eye movement dynamics. In *CVPR*, 2019.

[30] Wikipedia contributors. Optical flow — Wikipedia, the free encyclopedia, 2019. [Online; accessed 20-January-2020].

[31] Erroll Wood and Andreas Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. pages 207–210, 03 2014.

[32] Donghyun Yoo and Myung Chung. A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. *Computer Vision and Image Understanding*, 98:25–51, 04 2005.