

Development of a multi-platform library for the automatic detection of faces and eye centres in still images

Marc Oliu Simón

January 10, 2014

Director: Sergio Escalera Guerrero

Ponent: Cecilio Angulo Bahón

Titulació: Enginyeria Informàtica

Centre: Facultat d'Informàtica de Barcelona (FIB)

Universitat: Universitat Politècnica de Catalunya

Abstract

In recent years there has been an increase in the amount of content uploaded by the users to pages the goal of which is the management of contents, instead of their generation and distribution. There has also been a qualitative jump in the connectivity and ubiquity of information technologies, which has made possible for a huge amount of images to be generated distributively. Some examples of this trend are the popularization of the integrated cameras in mobile devices and tablets, the development of new smart digital cameras connected to the internet, the popularization of social networks and on-line storage.

This kind of technologies blurs the line between the generation of content and its management, and makes it necessary the implementation of a new set of automations for its processing. It is in this field where the smart algorithms for the automatic processing of images is very useful. In this project one of these problems is analysed, developed and implemented: the detection of faces and location of the eye centres in still images. The system has been designed in order to analyse the quality of portrait images, but the potential direct applications include content analysis, person counting, expression and person recognition applications, among others.

The developed method consists on an heuristic approach for discarding invalid face and eye detections working over the Viola & Jones object detection [38] algorithm to detect the faces and eyes bounding boxes. Afterwards, the eye centres are detected by using the fast radial symmetry transform [10] algorithm inside each eye bounding box. This kind of approach is fast because of the intrinsic speed of the used algorithms, and by using the specific problem information to discard incorrect detections, it is possible to relax the Viola & Jones detectors, detecting more correct faces in exchange of letting many incorrect ones to be detected. In that sense, information such as the intra-ocular distance or the eyes cross-correlation, along with the maximization of the face probability given multiple candidates, is used to determine if a face configuration is valid and select between multiple detected candidates.

It is the usage of the problem information that greatly improves the base accuracy of combining both methods, yielding much better results than those of some commercial solutions. In that sense, the algorithm is compared against the *Neurotechnologies* software, with the developed method obtaining better results.

Contents

1	Introduction	5
2	State of the art on facial analysis	6
2.1	Face detection	6
2.2	Data extraction and representation	7
2.3	Facial expression recognition	8
2.4	Multi-modal recognition of emotional states	9
3	Development of the methodology	10
3.1	The Viola and Jones object detector	11
3.2	The fast radial symmetry transform	12
3.3	Faces and eyes detection	13
3.4	False positives discarding	13
3.5	Eyes centre location	15
3.6	Face detection probability estimation	17
3.7	Eyes centre detection probability estimation	19
3.8	Face bounding box optimization	21
3.9	Error handling	21
4	Library design	22
4.1	Use cases	22
4.2	Classes design	23
4.3	Sequence diagrams	28
4.3.1	Library initialization	28
4.3.2	Parameters definition	28
4.3.3	Faces and eye centres detection	28
4.3.4	Results elimination	30
4.3.5	Error message obtention	32
5	Experimental methodology	32
5.1	The tested dataset	32
5.2	Accuracy analysis	33
5.3	Temporal cost analysis	35

6 Experiments	35
6.1 Parameter selection	35
6.1.1 Minimum face heterogeneity	35
6.1.2 Maximum eyes pair cross-correlation	36
6.1.3 Allowed distance between eyes	37
6.1.4 Radial symmetry test radii	38
6.1.5 Eyes binary mask threshold	39
6.1.6 Fast radial symmetry degree	40
6.1.7 Selected parameter values	41
6.2 Face detection accuracy	41
6.3 Error margin versus Recall	43
6.4 Precision versus Recall	44
6.5 Comparison with other solutions	45
6.6 Temporal cost	47
7 Conclusions	48
8 Future work	49
A Accuracy tables for the face heterogeneity parameter selection	53
B Accuracy tables for the eyes heterogeneity parameter selection	54
C Accuracy tables for the eyes min/max distance parameters selection	55
D Accuracy tables for the eyes symmetry min/max test radii parameters selection	57
E Accuracy table for the eyes binary mask threshold parameter selection	62
F Accuracy table for the fast radial symmetry degree parameter selection	63

1 Introduction

In this project an algorithm for the automated detection of faces and the eye centres is developed and implemented which intends to replace the commercial solution presented by Neurotechnologies. This algorithm allows for the detection of these features in still images, as well as the estimation of the probability of the detected features being correct.

This kind of task uses to be developed as a previous step to the extraction of information or image processing, and for this reason first an analysis of the problem inside a bigger context is performed. In the current case, the solution has been implemented as part of a software for the automated analysis of portrait images, where the goal is to obtain a correct framing for the image, as well as to determine its quality by evaluating the face frontality and inclination, illumination, presence of shadows, face obstructions and background homogeneity, between other image features.

The developed methodology is based on the Viola & Jones [38] object detection algorithm to detect the faces and eyes centres, and uses an heuristic approach in order to discard false positives from the faces, which allows the Viola & Jones algorithm to be relaxed, detecting more true positives in exchange of detecting many more false positives. The heuristic approach also evaluates all possible face configuration probabilities when more than two eyes are acceptable for a given face, keeping the eyes pair that maximizes the face probability. With the faces and eyes bounding boxes detected and the false positives discarded, the eyes centres are located by using the fast radial symmetry transform algorithm [10]. This algorithm is used in order to detect the centre of the iris radial symmetry, and is combined with a binary mask in order to discard false symmetries from shadows present in the eye bounding box. Finally, an estimation of the face and eye centres probabilities is performed by modelling some face configuration and eye centre features into two multivariate normal distributions.

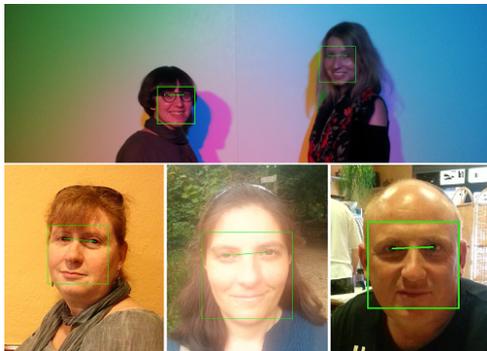


Figure 1: Sample processed images showing the features detected by the methodology.

In the first part of this document an analysis of the general methodology for the facial analysis is performed, analysing specific methods that currently exist at each step of the process. This is done by evaluating a widely explored problem which includes face detection and face features subtraction: face expression recognition. By analysing a broader well-studied problem, the main steps that both tasks have in common can be explored and used as a guideline to build the new methodology.

After analysing the currently existing methods for the different tasks, a custom methodology is developed at Section 3, defining a set of algorithms to use at each problem sub-step, as well as an heuristic approach to take into account the information specific to the problem of detecting faces and their eye centres. Once developed, the library design is given in Section 4 by means of a set of UML diagrams.

Finally, in Sections 5-6, an experimental methodology is proposed and executed consisting on a 10-fold cross validation over a dataset of 1000 frontal face images, where at each fold the algorithm parameters are optimized and then the algorithm accuracy, precision and temporal cost evaluated. A comparison is also performed between the proposed method and the commercial SDK developed

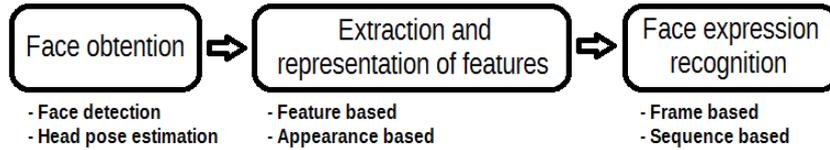


Figure 2: Diagram showing the general flow of an application for the analysis of facial expressions.

by Neurotechnologies this library intends to replace though the usage of an independent testing dataset of 266 images, which shows that the new method outperforms the commercial one in terms of the recognition rate.

2 State of the art on facial analysis

In this section it is intended to perform an analysis of the general process currently used for recognizing facial expressions, as well as categorizing the types of methods that are used in the involved steps. This study is performed in order to analyse the best options for developing a custom method for the detection of faces and extraction of its features.

The general structure of a face detection and analysis system can be divided into three steps [47, 1], as it can be shown in Figure 2. First, the acquisition of the face(s) present in the image is required, process that consists on its detection and posterior analysis of its orientation. Then a set of features is subtracted, defining the state of the face. These features fall into one of two general groups: based on the general analysis of the face appearance, or by subtracting a set of specific features defining a state for certain regions.

Finally, the obtained features are analysed in order to determine which type of expression is the face showing. This again can be done with two different general approaches. The first one consists on analysing the features of a single image, while the other puts image sequences in context in order to analyse the expression in a temporal context.

2.1 Face detection

Face detection on images is a specific case of the object detection methods, that have already been studied in depth, and for which a group of highly efficient or flexible methods exists, depending on the requirements of the application.

The **image based methods** initially were the most used algorithms for the detection of faces, consisting in the used of conventional classifiers, where the information of the pixel intensity of an image region is used as parameters of the features vector, classifying between *face* and *non-face* classes. Following this approach, methods based on Support Vector Machines (SVM) [7] and Artificial Neural Networks (ANN) [14] analyse difference image regions at different scales at a specific resolution, using the image pixels as features for the feature vector.

There are other image based methods that do not use a classifier in order to determine if a region displays a face. This is the case for the method developed by K.K. Sung and T. Poggio [25], which uses a set of training faces to manually determine the number of clusters that are formed in the pixel space, afterwards obtaining the mean face for each group and calculating its distribution. When a new face is presented, the probability of it belonging to a *face* cluster is obtained from the distances between the face and the mean faces.

The **feature based methods**, on the other hand, obtain a set of features from the image by applying mathematical operations on it. This allows in many cases to obtain systems that

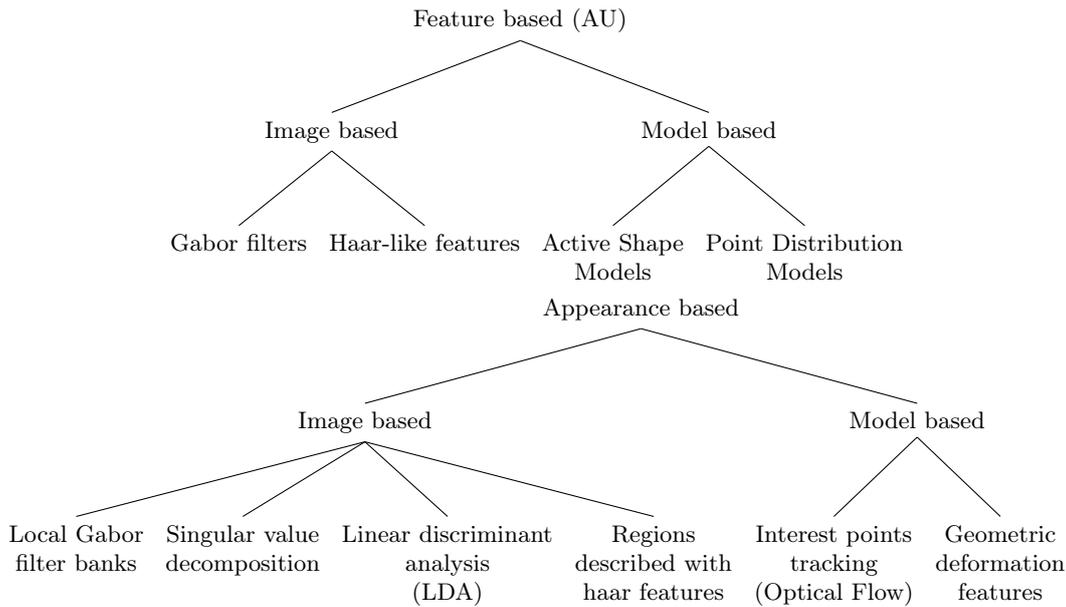


Figure 3: Classification of the facial feature extraction and representation methods for the feature based and appearance based methods.

work much faster and with equal or even higher accuracy, since many variations at the pixel level are compensated when basing the features on the measure of symmetries and asymmetries of the faces. This is the case for methods based on the object detection algorithm developed by P. Viola and M.J. Jones [34, 39, 40].

Although both method families use to implement mechanisms to obtain invariance to scale and illumination, they don't use to be invariant to rotation. This means that when the orientation of the faces is unknown, the images must be analysed for a wide set of orientations, multiplying the computational cost. In order to solve this problem, some methods implement mechanisms for the automatic recognition of a region orientation as a previous step to face detection. An example of this type of methods it the one proposed by H.A. Rowley *et al.* [15].

Computationally speaking, the most efficient method is the one proposed by P. Viola and M.J. Jones [39], commonly known as the Viola & Jones object detection method. Since it works directly with the integral image in order to calculate the image characteristics, it is unnecessary to rescale the image to calculate features at different scales, and reducing the computational cost of calculating a feature to an arithmetic operation between region extremes. This method is explained in greater detail on Section 3.1, at the development phase of the methodology.

2.2 Data extraction and representation

The second step in the process, that is, the extraction and representation of the data, can be performed in many different ways, as it is shown in the classification on Figure 3. These methods can be divided into two big groups, the ones *based on the appearance* and the ones *based on features* of the face.

The **feature based** methods usually make use of the position and shape of the face components (including eyes, mouth, eyebrows and nose) that characterize the facial expressions. This type of features are defined under the standard FACS (Facial Action Coding System) [41], and are called Action Units (AU). The description of the AU can be performed by means of two general types of methods: The image based ones and the model based ones.

The image based methods try to describe the AU by means of feature vectors that define the specific image region from its graphic features. In this type of methods there are the local Gabor filter banks [16] and the haar-like features [49].

The other group of methods is based on the description of the AU by fitting a previously trained or manually described model to the image. This can be done by means of the location of discrete features, like the Point Distribution Model (PDM) [4], or modelling the region by means of a geometric model, like in the case of the Active Contour Model [32].

The **appearance based models** define the described element as a single block instead of analysing interest regions. This type of model can be divided into two groups like in the previous case.

The image based methods describe the whole face from a group of general descriptors, in a manner similar to how the regions corresponding to AUs were analysed in the case of feature based models. In this group there is the local Gabor filters bank [16], the Linear Discriminative Analysis (LDA) [5], the singular values decomposition, the singular values decomposition (SVD) [5] and the usage of haar-like features [22, 49].

In the case of model based methods, there are those methods that analyse random points of the face, normally generating a model that defines the general structure of the face itself. In this type of methods fall the interest points tracking [44] method, which consists on analysing the optical flow of an expression over a sequence of images, and the geometric deformation features [18] method, which consists on analysing the geometric deformation of a model fit to the analysed face with respect to the same model over the normalized face.

2.3 Facial expression recognition

Once obtained the feature vectors for the image or images sequence to analyse, these vectors are used in a classification method that allows to identify the facial expression. These methods can be grouped into two general categories. By one side there are the *frame based methods*, which work over the feature vector of a single image to identify the expression. On the other hand, the *sequence based methods* analyse the feature vectors of multiple contiguous frames in order to take into account the temporal information when identifying the expression. Figure 4 shows a classification of the existing types of methods for facial expression recognition.

The **frame based methods** are the simplest kind of methods. These are based in the classification of a feature vector obtained from a single frame, and direct methods can be used to classify the vector, like discriminative neural networks [29, 32, 5] or a direct comparison of the vector with a prototype vector of the different facial expressions over a previously defined comparison space [33].

Another group of methods implies the usage of a set of conventional binary classifiers assembled by means of ECOC [45]. Some usage examples of this type of classifiers for classifying facial expressions are the SVM and AdaBoost [36, 22, 49].

The **sequence based methods**, on the other hand, have into account the information of a set of frames in temporal order, such that information about the expressions evolution over time is taken into account when performing a classification. This type of methods can be classified into two general groups.

By one side, it is possible to work over a set of static descriptors, each one of which contains the information obtained in a single frame. This type of classification methods use techniques like Dynamic Bayesian Networks (DBN) [48] or the Hidden Markov Models (HMM) [42] in order to establish a temporal relation between the frames, calculating the probability of a frame sequence taking into account the probability of a frame given the previous ones.

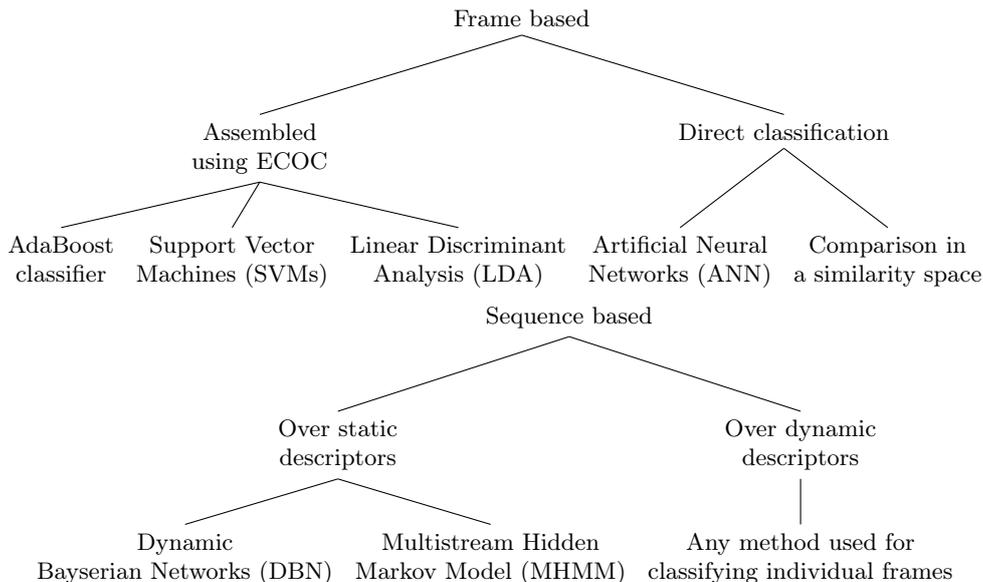


Figure 4: Classification of the facial expression recognition methods for the frame based methods and the sequence based methods.

By the other side, if the used descriptor implicitly contains the temporal information, like the deformation of a set of features, the same set of methods used to analyse individual frames can be used over the new feature set. Some examples of this type of approaches can be seen in [35].

2.4 Multi-modal recognition of emotional states

Facial expression recognition is not the only technique used for the automatic recognition of emotional states. In face, another very popular technique comes from voice analysis [46, 3], in which different voice features are analysed over time, normally using probabilistic graphical models, like the Hidden Markov Models.

Other studies have focused in the analysis of gestures, like the one seen in [6], or the physiological information like the pulse, electrical conductivity of the skin, blood pressure, dilation of the pupil, etc. Two examples of this approach are [23, 20].

For facial expression analysis, though, the techniques that better complement the available information are those analysing the 3D information of the face [21, 12, 13] and its thermal images [26, 2], since these data modalities have a spatial correspondence, as well as a temporal one in case of sequences. This would allow the data sources to be aligned and better fused the information modalities in the feature vector.

Even though many of these techniques have been individually studied in depth, it hasn't been until recently that multi-modal techniques have been implemented for emotion recognition. The first ones to be developed, as well as the most obvious, were the ones based on the analysis of audiovisual material, analysing both facial expressions and voices together [28, 30, 31, 37].

More recently bimodal techniques based in the analysis of facial expressions and gestures have appeared [11], as well as for the voice and physiological state [24, 19], and multi-modal techniques that combine the facial expressions and the voice with thermal images [50] or gesture information [9].

In order to jointly analyse data from multiple modalities, there are different types of techniques, as it can be seen in the scheme on Figure 5. These are classified into two general groups depending

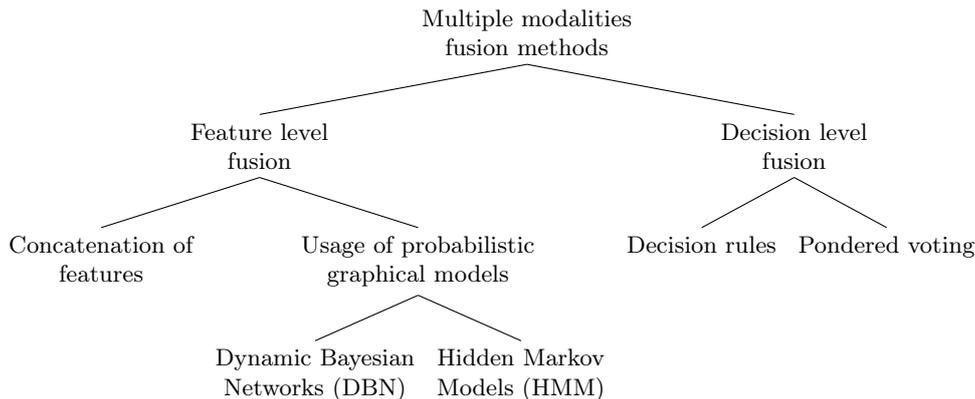


Figure 5: Classification of the fusion methods for multi-modal emotion recognition.

on the data being fused at the features level or at the decision level.

Data fusion at the decision level consists on first applying emotion recognition separately with each data modality, and then giving preference to one of the two results with a rule based system [28, 30] or with a pondered voting with previously established weights [11, 24, 19, 50, 43].

Feature level fusion consists on directly fusion the modalities at the data analysis stage. This can be done by means of a *direct features concatenation*, where a classifier that uses the most relevant features of each modality along the sequence is built [31, 24, 43], or *using probabilistic graphical models* like a Dynamic Bayesian Network [37] in order to take into account the probabilities of each modality in each step of the sequence with relation to the data and the probabilities in the previous steps. This second group of methods usually obtain a higher mean correct classification rate.

One of the fields that has not been given much attention is the usage of three-dimensional information of the face in combination with information of other modalities. One of the reasons for which these modalities have not been given much attention is that the information they offer is only useful to increase the precision of the analysis of the facial expression, but does not add complementary information about the emotional state.

Even though, a system that combines two-dimensional information with its corresponding three-dimensional information to analyse facial expressions in sequences has been developed . It has been presented by L. Yin *et al.* [27] as a validation method of a new database with 2D and 3D information for expression sequences. This method uses a two-dimensional Hidden Markov Network, with a state for each frame of the sequence. This means that the information of the two modalities is analysed jointly, and not through a linear combination of classifiers.

3 Development of the methodology

The developed methodology consists on multiple steps, which are illustrated in Figure 6. First the face detection process is performed with the Viola & Jones face detection algorithm (Section 3.3), since it has been shown in Section 2.1 to be the fastest object detection method. Afterwards, the same technique is applied for eyes detection inside the face regions, thus reducing the search space for the location of the eyes centre.

In order to discard the eyes bounding box false positives inside of each face region, first a series of sanity checks are performed (Section 3.4), afterwards evaluating all possible face configurations (Section 3.6) and selecting the one maximising the probability given a pair of eyes. The rest of face configurations are considered as false positives. This false positive filtering approach allows

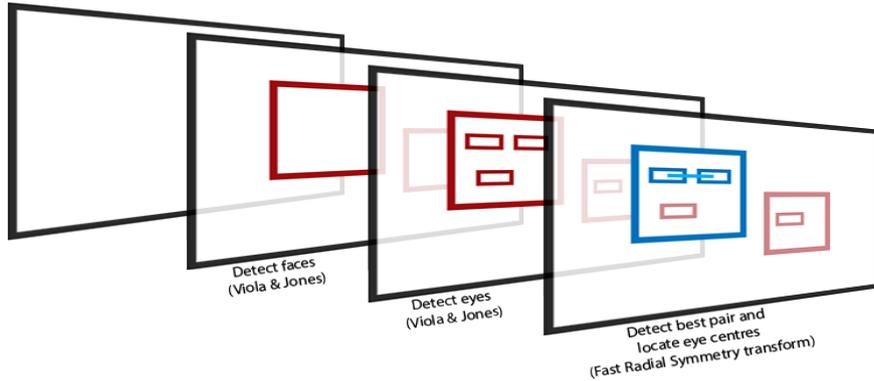


Figure 6: General steps of the methodology.

for the Viola & Jones detectors to be relaxed, increasing the detection rate over the True Positives in exchange of detecting many false positives.

With the eyes bounding boxes detected, it is now possible to recalculate the face bounding box in order for it to match the required standard, as it is explained in Section 3.8. This is necessary due to the Viola & Jones technique detecting face symmetries that do not necessarily require the eyes to be in an exact position inside of the face rectangle, leaving a non-centred framing. With the eyes bounding boxes located, it is possible to correct this problem.

Once the optimal pair of eyes is selected, the centres of the eyes are located. This is done by first scaling each eye bounding box to a fixed size and applying a methodology based on the fast radial symmetry transform (Section 3.5). Finally, the evaluation of the eyes pair centres is performed by means of the same technique used for face configurations (Section 3.7).

3.1 The Viola and Jones object detector

The Viola & Jones object detector [38] used for the detection of faces and eyes consists on a cascade classifier that uses Haar-like features. This kind of detectors define a set of features, which are measured over face candidates and used in a binary cascade classifier in order to decide if the face candidate belongs to the object or non-object class. This process is performed over a sliding window that moves through the whole image at different scales.

The sliding window: In order to detect all of the objects inside of an image, with this type of method all of the possible candidates have to be analysed. This is performed by using a window of fixed proportions that is displaced through all possible positions of the image for a wide range of scales, between a defined minimum and maximum equal to the image size. For each scale and position of the sliding window, the image segment falling inside of the window is a candidate to the object being detected.

The Haar-like features: For each candidate a set of real-valued features is subtracted. Each feature corresponds to the difference between the sum of pixel intensities from the positive and negative regions of the candidate. The positive and negative regions of a feature are defined by a mask, as it can be seen in Figure 7. Using rectangular regions and the integral image, the sum of pixels of a regions can be rapidly calculated by using the values at the extremes of the region, according to the formula $area = img_{br} + img_{ul} - img_{ur} - img_{bl}$.

The cascade classifier: In order to speed up the detection process, a cascade classifier is used, which at each step uses an AdaBoost classifier using a subset of the features. The AdaBoost classifier is trained in order to discard as many candidate regions as possible without discarding true positives, such that at each step a set of candidates is discarded. By using this method,

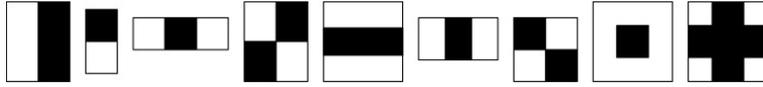


Figure 7: Example of a set of Haar-like features. The white regions indicate positive regions of the image, while the black ones are negative regions.

only a small subset of the features needs to be calculated in most cases, discarding most non-face regions at the first steps of the cascade classifier.

This type of classifier returns a rectangular region of the same proportions as the detection window, and in the developed methodology is used first for face detection, and then inside the corresponding regions in order to detect the bounding box for the eyes.

3.2 The fast radial symmetry transform

The fast radial symmetry transform [10] is a technique developed by G. Loy and A. Zelinsky in 2002 that allows for the detection of radial symmetries in images in a fast manner. This algorithm consists on a series of steps that take into account the gradient direction at each image position, as well as its intensity, in order to cast a positive vote in the positive direction of the gradient, and a negative vote in the negative direction, as it is illustrated in Figure 8.

The specific steps followed by the algorithm are the following ones:

Step 1: Generate convolutions

The first step of the process is to generate a vertical and an horizontal convolution image. These images are generated from a mask that performs a Gaussian smoothing in the direction opposite to the edge detection direction, and a Laplace transform in the other direction, effectively performing a vertical/horizontal smoothed edge detection.

Step 2: Calculate intensity map

An intensity map of the gradients is performed, where for each pixel the vertical and horizontal convolution intensity are linearly combined to obtain the module of the linear combination. After calculating this value for each pixel of the image, this gradient intensity map is normalized.

Step 3: Generate radial symmetry map

Step 1: Create a real-valued matrix of the same dimensions of the original one with all values set to zero.

Step 2: For each pixel of the original image, the direction of the gradient is obtained by using the value for that pixel at both convoluted images as vector modules.

Step 3: Following the direction of the gradient, the values on the symmetry map at a distance equal to the evaluated symmetry radius are voted positively in the positive direction of the gradient, and negatively at the opposite direction. The value added to these positions is defined at Equation 1.

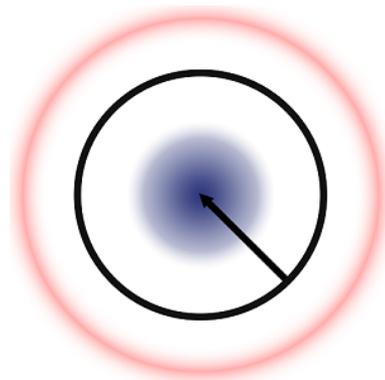


Figure 8: The fast radial symmetry transform algorithm votes positively in the gradient direction (blue), and negatively in the opposite direction (red).

The obtained symmetry map represents the level of radial symmetry of each pixel at the evaluated radius. As seen in Equation 1, the degree of symmetry represented at the position is

defined by the α parameter, with a higher value representing a higher degree of symmetry. The kappa parameter, on the other hand, is a constant that has empirically been found to have an optimal value of 8.0 for a radius value equal or lower than 1, and of 9.9 for higher radius values.

$$\begin{aligned} dir_o &= (symm_d < symm_o)? - 1 : 1 \\ symm_d &= symm_d + \frac{inte_o}{kappa} \cdot \left(\frac{dir_o}{kappa} \right)^\alpha \end{aligned} \quad (1)$$

3.3 Faces and eyes detection

Face detection is a procedure that is performed in multiple steps and is slightly related to the eyes detection. First, all face candidates in the image are detected by means of a Viola & Jones object detector, explained in Section 3.1. Then, the same methodology is used in order to detect the eye candidates inside of the face regions. The eyes bounding boxes detection is performed in the first place to reduce the execution time in posterior steps of the method, but they are also used in order to discard faces false positives.

Once the eye candidates bounding boxes are located for a given face, a procedure is performed in order to detect the best eyes pair, since false positives could have been detected. This procedure consists on first processing the eye candidates in the following manner:

1. Remove candidates located at the lower half of the face
2. Remove candidates located too close to the face bounding box
3. Classify candidates according to their geometric centre being at the left or right half of the face

Afterwards, all possible combinations of left and right eye candidates are evaluated, also following a multi-step methodology explained below:

1. For each candidate pair
 - (a) Discard if eyes too close or far apart
 - (b) Evaluate face probability given the eyes pair
2. Select pair with the highest score
3. Discard all other candidates

3.4 False positives discarding

Once detected the bounding boxes for the faces and eyes candidates, the next step is to discard the false positives. This can be done by defining a set of rules that must be fulfilled by the detected bounding boxes in order to be considered correct, discarding all the others. The pseudo-code for the methodology followed for discarding false positives is shown in Figure 9.

First, all faces with an intensity heterogeneity lower than a given threshold are discarded. This discards the false positives corresponding to mostly homogeneous areas that thanks to small illumination fluctuations fulfil the symmetry criteria required by the Viola & Jones detector. The rest of true positive candidates are evaluated by using the face configuration defined by the face candidate and its eyes, which means that first the false positives for the eyes must be discarded.

```

for all face  $\in$  candidates do
  if heterogeneity(face.area) < HETEROGENEITY_THRESHOLD then
    candidates.remove(face)
    continue
  end if

  eyesLeft  $\leftarrow$  []
  eyesRight  $\leftarrow$  []
  for all eye  $\in$  face.eyes do
    if eye.centresY > face.centresY then
      continue
    end if
    if eye.centresX < face.centresX then
      eyesLeft  $\leftarrow$  eye
    else
      eyesRight  $\leftarrow$  eye
    end if
  end for

  if |eyesLeft| = 0 and |eyesRight| > 0 then
    eyesLeft  $\leftarrow$  eyesRight
    eyesLeft.left = eyesRight.right
    eyesLeft.right = eyesRight.left
  else if |eyesRight| = 0 and |eyesLeft| > 0 then
    eyesRight  $\leftarrow$  eyesLeft
    eyesRight.left = eyesLeft.right
    eyesRight.right = eyesLeft.left
  end if

  pairs  $\leftarrow$   $\forall$ [left, right]|left  $\in$  eyesLeft, right  $\in$  eyesRight
  bestPair  $\leftarrow$  NULL
  for all pair  $\in$  pairs do
    dist  $\leftarrow$  eyesDistance(pair.left)
    if dist > MIN_DISTANCE and dist < MAX_DISTANCE then
      pair.prob  $\leftarrow$  probability(face, pair)
      if pair.prob > bestPair.prob then
        bestPair  $\leftarrow$  pair
      end if
    end if
  end for

  if bestPair = NULL then
    candidates.remove(face)
    continue
  end if

  face.eyes  $\leftarrow$  {bestPair.left, bestPair.right}
end for

```

Figure 9: Pseudo-code describing the developed methodology for discarding face and eye false positives and assigning the pair of eyes bounding boxes maximizing the face probability to each face.

In order to discard the eyes false positives for a given face, first the eye candidates found at the lower half of the face bounding box are discarded. Afterwards, the eyes are classified between left and right eye candidates, and all possible pairs of left and right candidates are evaluated by using the probability of the face configuration given a pair of eyes. If there are no left or right eye candidates, the present candidates detected at one side of the face are projected to the other side, and accepted as candidate eyes as long as the difference in pixel intensities between the original and the projected eye bounding boxes is lower than a given threshold. Since a face can only have two eyes, the problem is now reduced to determining which one of the pairs is the best one.

All of the eye pairs for which the distance between the eyes bounding box centres are too close or too far away are discarded, and the remaining pairs are evaluated by calculating the probability of the face configuration for a given pair as explained in Section 3.6. The pair of eye candidates maximizing the face probability is taken as valid, discarding the others. If there is no valid pair of eyes remaining for the evaluation, the whole face candidate is discarded.

In order to perform the explained procedure, a series of constants defining certain restrictions imposed to the face and eye candidates have been found. These constants must be empirically determined, as it is done in Section 6.1, and are the following ones:

- Minimum face intensity heterogeneity
- Minimum eyes pair cross-correlation
- Minimum and maximum intra-ocular distance

3.5 Eyes centre location

In order to locate the eye centres, an approach that uses the fast radial symmetry transform method explained in Section 3.2 is used. the pseudo-code for the specific algorithm is shown in Figure 10, with the general steps of the procedure being the following ones:

Step 1: Scaling

The bounding box of the eye is scaled to a fixed size of 100x60 pixels in order to reduce the computation time and use the same test radii.

Step 2: Radius estimation

The fast radial symmetry transform technique is applied for radii values between a minimum and a maximum defined in the optimization step at Section 6.1.4. Only the positive gradient direction is taken into account, and the selected optimal radius is the one that has the highest symmetry value in any of the image pixels of its symmetry map.

Step 3: Obtention of the symmetry map

The symmetry map for the optimal radius is merged with the symmetry maps for the immediately inferior and superior radii. This compensates for the pixelling of the image and the imperfection of the circumference due to a perspective distortion.

Step 4: Centre selection

The position of the pixel with the highest symmetry value of the symmetry map is the position of the eye centre.

Since an image rescaling is performed at the first step of the process, the eye iris has a similar size in all cases, independently of the original resolution of the image. This allows by one side to estimate the approximate radius of the iris (between 5 and 10 pixels), and by the other to estimate the distortion at the iris edge because of pixelated images (about a pixel around the optimal radius). In this manner, all of the parameters for the algorithm can be determined with invariance to the processed image. These parameters are empirically determined in Section 6.1, and are the following ones:

```

Require:  $face \neq \emptyset$ 
for all  $eye \in face.eyes$  do
   $rato \leftarrow 100/eye.width$ 
   $eye_{bb} \leftarrow scale(eye.boundingbox, rato)$ 
   $mask \leftarrow binaryMask(eye_{bb}, MASK\_THRESHOLD)$ 
   $symmaps \leftarrow \mathbf{new} Map[RADIUS\_MIN .. RADIUS\_MAX]$ 

   $maxrad \leftarrow \{value \leftarrow 0, map \leftarrow 0\}$ 
  for all  $radius \in [RADIUS\_MIN .. RADIUS\_MAX]$  do
     $symmaps_{radius} \leftarrow symmetryTransform(eye_{bb}, radius, SYMMETRY\_DEGREE)$ 
     $symmaps_{radius} \leftarrow applyMask(symmaps_{radius}, mask)$ 
    if  $max(symmaps_{radius} > maxrad.value)$  then
       $maxrad \leftarrow \{value \leftarrow max(symmaps_{radius}), map \leftarrow radius\}$ 
    end if
  end for

   $symmap \leftarrow \mathbf{new} Map$ 
  for all  $radius \in [maxrad.map - 1 .. maxrad.map + 1]$  do
     $symmap \leftarrow symmap + symmaps_{radius}$ 
  end for

   $[intensity, x, y] \leftarrow max(symmap)$ 
   $eye.centre.x \leftarrow eye.left + (x - eye.left) \cdot rato^{-1}$ 
   $eye.centre.y \leftarrow eye.top + (y - eye.top) \cdot rato^{-1}$ 
end for

```

Figure 10: Pseudocode describing the developed methodology for locating the eyes centre by generating the symmetry maps for a range of test radii, afterwards merging the best one with its neighbors.

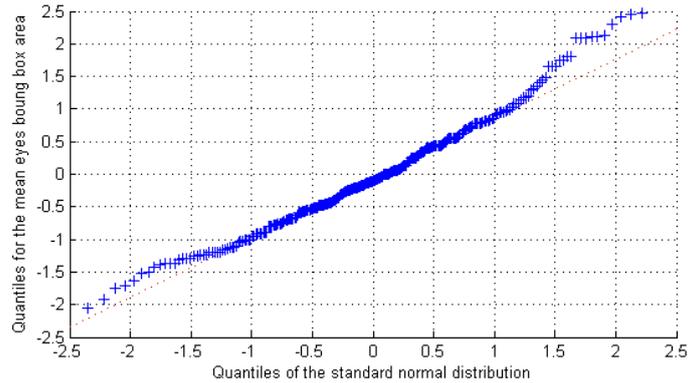


Figure 11: Quantile-Quantile plot comparing the distribution of the eyes mean surface samples with the standard normal distribution.

- Radial symmetry test radii
- Eyes binary mask threshold
- Fast radial symmetry degree

3.6 Face detection probability estimation

In order to estimate the probability of a face being correctly detected, a different method from the one used to detect them in the first place must be used. This is due to the face and eyes detection algorithm, which even though it is very efficient computationally, is a binary classification tool that does not offer a certainty margin for the obtained detections.

The used criterion in order to estimate the probability of the detected face really being a face is the analysis of a set of features of the eyes bounding box distribution inside of the face bounding box. More precisely, the three following features have been used:

Relative mean eyes area: This feature describes the mean area of the eyes bounding boxes for a given face, relative to the area of the face bounding box. Since the bounding boxes for the detections are always orthogonal with respect to the image, the relation between the face and the eyes in an image changes with the face inclination.

Relative distance between the eyes: It is the euclidean distance between the centres of the eyes bounding boxes divided by the width of the face bounding box. This distance is an estimation of the separation between the eyes that is invariant to the face size.

Relative mean eyes height: It is the distance from the upper part of the face bounding box to the mean height of the eyes bounding box centres, divided by the height of the face bounding box. This value indicates where the eyes are located vertically in the face.

These three features fulfil the condition of being invariant in relation to the orientation of the face, such that the developed statistic cannot suffer from a significant bias in the selected sample due to a preferred orientation of the faces. Furthermore, all of the features follow a normal distribution for the vast majority of the samples, as it can be seen in the plots comparing the normal distribution against the distributions followed by these features by means of a Quantile-Quantile plot, which can be seen in Figures 11-13.

As it can be seen from the plots, even though the distribution doesn't strictly follow a normal distribution, because of the small number of instances of small probability (outside of the two first quantiles), it does follow a normal distribution for the vast majority of the distribution. For that

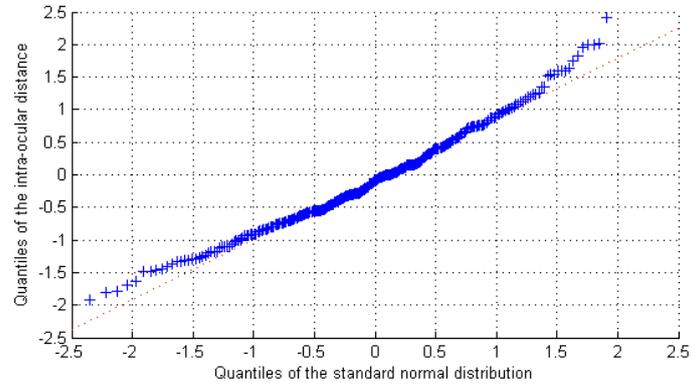


Figure 12: Quantile-Quantile plot comparing the distribution of the intra-ocular distance samples with the standard normal distribution.

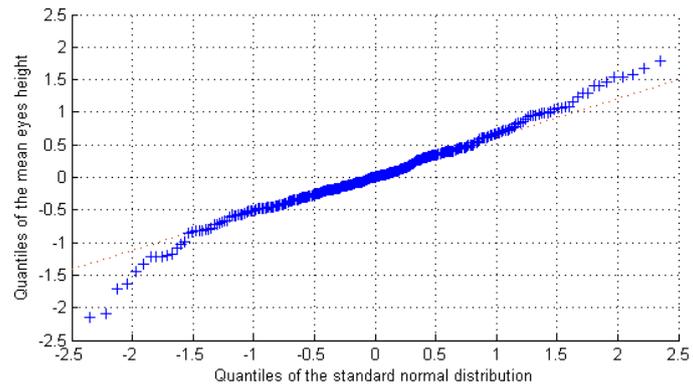


Figure 13: Quantile-Quantile plot comparing the distribution of the eyes mean height samples with the standard normal distribution.

reason, considering that the individual features follow a normal distribution, a three-dimensional multivariate normal distribution has been generated by using each one of the individual features as a dimension of the problem.

A multivariate normal distribution is defined as $\mathbf{x} \sim N_k(\mu, \Sigma)$, where μ is a vector of length k containing the mean for each dimension, and Σ represents the covariance matrix of $k \cdot k$ dimensions. In order to estimate the μ and Σ values of the distribution, a sampling of 1000 instances has been used for which the ground truth is known. The obtained sample means μ can be seen in the following table:

Mean eyes area	3.834430
Distance between the eyes bounding boxes	0.382549
Mean eyes vertical distance	0.387347

Table 1: Means vector for the faces multivariate normal distribution.

The covariance matrix for that multivariate distribution is the following one:

	Feature 1	Feature 2	Feature 3
Feature 1	1.318857	0.004796	0.000298
Feature 2	0.004796	0.001377	-0.000270
Feature 3	0.000298	-0.000270	0.000582

Table 2: Covariance matrix for the faces multivariate normal distribution.

In order to determine the probability of a new instance being a face, the *inverse cumulative* probability of that instance is calculated over the multivariate normal distribution. That is, the probability of a random distribution instance having a feature vector that is as much or less probable than the measured ones. While the calculation of the cumulative distribution can be easily calculated for a simple normal distribution, in multivariate normal distributions the precise calculation of this value involves the calculation of the integral area bounded by the sigma values for the given sample value at each dimension, which is a complex calculation that is performed by approximation.

This process is simplified by comparing the probability density of the evaluated instance against a set of random samples drawn from the distribution, determining the proportion of them with a higher probability density than the current one. By generating the random samples and calculating their *probability density* when first initializing the random distribution, the *inverse cumulative distribution* estimation is reduced to calculating the ratio of samples with a higher values.

3.7 Eyes centre detection probability estimation

In order to estimate the detection probability of the eyes centre, a procedure similar to the one seen in Section 3.6 for evaluating the faces probability is used. In this case, though, the eyes pair is evaluated jointly instead of individually. This is due to the fact that while in the case of faces the internal features of these can be used, for the eyes the available features only make sense when compared against the same features of the other eye. In this way, the position of one eye centre inside of its bounding box can be any, but must be approximately the same one the other eye has.

In order to evaluate this correspondence in the positions given an eyes pair, a multivariate normal distribution over a two-dimensional space, were the used features are the difference in the vertical displacement and the difference in the horizontal displacement, which are now explained in more detail:

Vertical displacement difference: It is the difference between the right eye vertical position

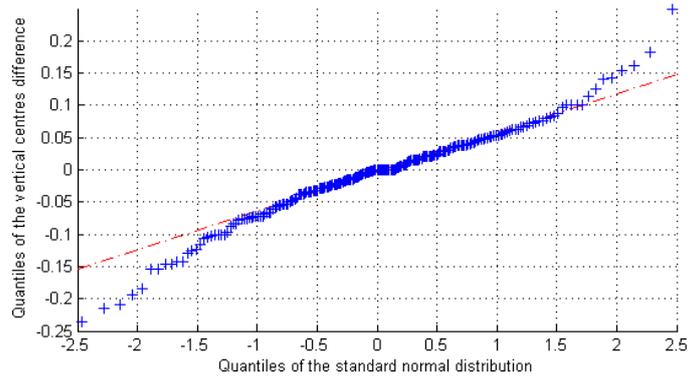


Figure 14: Quantile-Quantile plot comparing the sample distribution of the difference in the vertical direction of the eyes pair centres against the standard normal distribution.

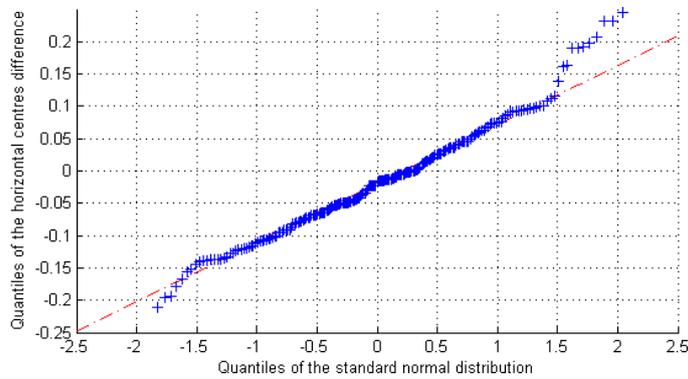


Figure 15: Quantile-Quantile plot comparing the sample distribution of the difference in the horizontal direction of the eyes pair centres against the standard normal distribution.

relative to its eye bounding box, and that of the right eye. The position in both cases is normalized by the height of the corresponding bounding box.

Horizontal displacement difference: It is the difference between the right eye horizontal position relative to its eye bounding box, and that of the right eye. The position in both cases is normalized by the width of the corresponding bounding box.

Both features, like in the case of the features used to calculate the faces probability, have been analysed over a sample of 1000 instances for which the ground truth is known in order to check that they follow a normal distribution by using the Quantile-Quantile plots, which can be seen in Figures 14 i 15. The plots show that the distribution of the population sample in both cases follows a normal distribution except for big sigma values, which means a normality presumption can be made for the studied features.

The bi-variant normal distribution $\mathbf{x} \sim N_2(\mu, \Sigma)$ which is derived of the used population sample has as mean μ the following values:

Vertical displacement difference	-0.019012
Horizontal displacement difference	-0.006474

Table 3: Means vector for the eyes multivariate normal distribution.

As for the covariance matrix of the Σ distribution, it takes the following values:

	Feature 1	Feature 2
Feature 1	0.011814	-0.000857
Feature 2	-0.000857	0.005455

Table 4: Covariance matrix for the eyes multivariate normal distribution.

3.8 Face bounding box optimization

While the Viola & Jones object detection algorithm gives an approximate position including the face itself when performing face detection, the face framing that is required of the algorithm is much more precise. The standard to be followed is a cropped version of the ISO/IEC 19794-5 standard [8], where the vertical bounds are located at the face sides leaving the ears outside, and the horizontal bounds are located under the chin and over the eyebrows respectively. This type of framing is illustrated in Figure 16.

The reason why the bounding box given by the Viola & Jones algorithm is just an approximation is that the algorithm works by detecting symmetries and asymmetries between regions, which yields imprecise detections. Also, the final detections are made by averaging multiple spatially close detections into a single one, which means the final bounding box is skewed in the direction where more detections are found, which is a problem when faces are not completely frontal.

In order to solve this, the positions of the eyes bounding box centres (not the eye centres themselves) are used in order to recalculate the bounding box. This can be done because of the eyes always maintaining a relation between their separation and the face width, and a precise vertical position. These proportions are empirically estimated from averaging the true proportions of the 1000 images dataset. The proportions found with this method are the following ones:

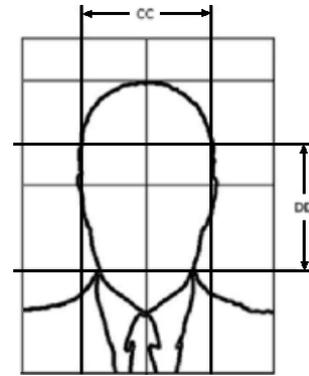


Figure 16: Bounds of the face framing to be detected by the algorithm.

- **Face width to eyes separation (2.11)** - Factor describing the distance between the face left and right bounds relative to the distance between the eyes.
- **Face height to eyes separation (2.12)** - Factor describing the distance between the face upper and lower bounds relative to the distance between the eyes.
- **Eyes mean height to eyes separation (-0.53)** - Factor describing the distance from the face upper bound to the mean eyes vertical position relative to the distance between the eyes.

3.9 Error handling

Another factor to take into account when designing the library is the management of special cases, where the image either is invalid or no results are found on it. In order to manage these cases, a series of checks are performed in different points of the methodology. The following list explains the cases and the detection procedure followed to determine them:

- *No error occurred* - If no error is found, that is, the image is bigger than the minimal size and one or more faces have been detected and accepted, no error is thrown.

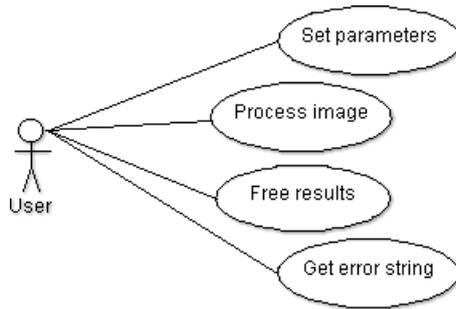


Figure 17: Use cases diagram for the developed library.

- *Image is too small* - This error occurs when the provided image is too small for the methodology to detect any face, even at the maximum possible resolution. This error is detected and thrown at the beginning of the algorithm by checking the original image size.
- *No faces found* - This error occurs when the Viola & Jones cascade classifier for the faces doesn't detect any face candidate. This type of error is checked after applying the face detection cascade classifier, and before detecting the eye candidates and discarding the false positives.
- *No eyes found* - This error occurs when no valid eye candidates are found, although there have been face candidates detected. This type of error is checked after performing face and eyes candidates detection and discarding the false positives, throwing the error if all face candidates are discarded.

4 Library design

In this chapter, a design for the library is developed based on the general methodology, specific techniques and modelled distributions seen during the development phase of the methodology in Section 3. The library design is realized by using the Unified Modelling Language (UML), first defining the general use cases the library must implement, and afterwards a class design to implement these use cases and the sequence diagrams specifying the execution flow of these use cases over the classes design.

4.1 Use cases

The use cases taken into account for the library design are the ones shown in the use cases diagram of Figure 17. As it can be seen, four use cases are taken into account, all of them interfacing with the host application.

The *set parameters* use case: Sets the values for the methodology parameters. While a set of experimentally chosen parameters are used by default, these can be modified in order to fine-tune the methodology to specific ethnicities, ages rank or sex.

The *process image* use case: Processes a given input image, detecting and scoring all of the faces and their corresponding eye centres. It also performs a small set of checks on the image in order to determine its validity, and returns the corresponding error code if required.

The *free results* use case: Frees a provided results structure previously obtained from the *process image* use case.

The *get error string* use case: Returns a string describing the error code obtained on the last usage of the *process image* use case. If no error occurred, a string stating so is returned.

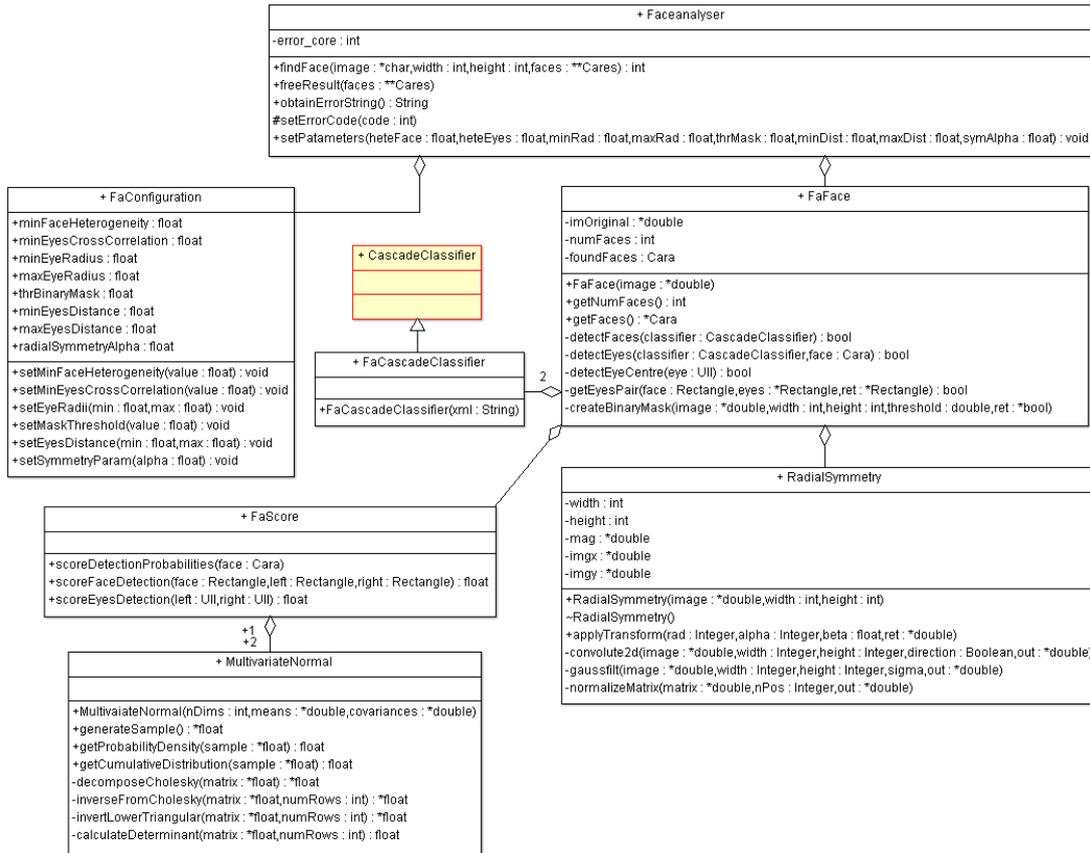


Figure 18: Classes diagram for the developed library.

4.2 Classes design

The developed library is internally structured by using object-oriented programming by following the scheme described in Figure 18, and uses an interface that creates a new instance of the *Faceanalyser* class in order to communicate with external applications. This type of structure allows for an easy reuse of the library source code in environments other than as an external library by removing the library interface and directly creating instances of the main class, and also allows for an easy parallelization of the library, since each *Faceanalyser* object creates its own collection of objects for the rest of classes it uses, not sharing any dynamic memory between instances.

The library classes structure is divided between a main class *Faceanalyser*, that implements the library calls, the *FaFace* class, implementing the principal methods for both face detection and eyes centre location, the *FaConfiguration* class, which serves as a storage for the different parameters, the *FaScore* function, which scores the probabilities for both the faces and eye centres, and the *RadialSymmetry* and *MultivariateNormal* classes, which implement the *fast radial symmetry transform* and the *multivariate normal distribution* methods respectively. These classes and their main methods are explained in detail in this section.

Faceanalyser class

This class implements the main methods of the library interface, and by instantiating it, their associated *FaFace* and *FaConfiguration* objects are instantiated and assigned to itself. When the library is loaded, a new *Faceanalyser* object is created, and all of the interface functions are

pointed to their corresponding *Faceanalyser* instance methods.

findFace(image, width, height, faces): errorCode

This method implements the main library call, in charge of detecting the faces and eye centres inside of *image* with size *width · height*. A pointer to a structure containing the detected components is stored inside the *faces* parameter, with the function returning zero when successful, and the corresponding error code if not.

*freeResult(faces: **Cares)*

This method is in charge of freeing the memory of the previously allocated structure containing the information on the detected faces, which was generated by the *findFace* method.

obtainErrorString(): errorString

This method returns an explanatory string for the error code obtained from the last call to the *findFace* method. In there was no error, a string stating so is returned.

setParameters(heteFace, heteEyes, minRad, maxRad, thrMask, minDist, maxDist, symAlpha)

This method sets the methodology parameters for both face detection and eyes centre location by calling the required setters of the *FaConfiguration* object associated to this *Faceanalyser* instance.

FaFace class

This class implements the general methodology for face and eyes detection with False Positives discarding, as well as the binary mask generation itself. The specific algorithms for the face and eyes bounding box detections as well as the algorithm for the eyes centre location, though, are found in other classes used by this one.

FaFace(parent, image)

This method corresponds to the class constructor, and its main function is to first store the *parent* class, which is the *Faceanalyser* object passed as parameter and is used to access the methodology configuration, and to then perform the face detection and eyes centre location procedure.

getNumFaces(): numFaces

This method returns the number of faces detected on the image.

getFaces(): faces

This method returns a vector of *Cara* structures containing the information on the faces detected over the image.

detectFaces(classifier): foundSomething

This method applies the *classifier* haar-like cascade classifier to the image, storing the detected face bounding box candidates into the object attributes along with the number of detected faces. It returns a boolean value indicating if any face has been found.

detectEyes(classifier, face): foundSomething

This method applies the *classifier* haar-like cascade classifier to the image region corresponding to the face passed by parameter. It stores the detected eyes bounding box candidates inside of the *face* structure that was passed by parameter, and returns a boolean indicating if any candidate has been found.

detectEyeCentre(eye): centre

This method is in charge of detecting the eye centre inside the bounding box of the eye passed by parameter. It does so by first determining the optimal eye radius and merging the neighbouring symmetry maps, as explained in Section 3.5. The *fast radial symmetry transform* algorithm is called from this function by creating a temporal object of the *RadialSymmetry* class. It returns the point coordinates of the detected eye centre.

getEyesPair(face, eyes, ret): foundPair

This method evaluates all accepted individual eyes by first discarding the ones found at the lower half of the face, and then evaluating all pairs of left and right eye candidates by following the criteria described in Section 3.4. The best pair, if any valid pair is found, is stored at the *ret* vector that was passed by parameter. The return value is a boolean indicating if a valid pair has been found.

createBinaryMask(image, width, height, threshold, mask)

This method creates a binary mask that is to be applied to the symmetry map when looking for the centre of an eye. The mask is created for the provided image *image*, with all intensity values under *threshold* being considered as positive mask values (for the normalized image). The resulting mask is stored at the *mask* buffer passed by parameter.

FaConfiguration class

This class stores the values for the different parameters used by the algorithm. A pre-defined set of experimentally chosen values are used to initialize the parameters, although these can be changed through a series of class methods. The methods for modifying these values, as well as the values set by each one, are the following ones:

- *setMinFaceHeterogeneity(value)* - sets the minimum face heterogeneity value.
- *setMinEyesCrossCorrelation(value)* - sets the minimum value for an eyes pair cross-correlation.
- *setEyeRadii(min, max)* - sets the minimum and maximum test radii when determining the optimal eye radius.
- *setMaskThreshold(value)* - sets the intensity threshold for generating the binary mask for eye centre location.
- *setEyesDistance(min,max)* - sets the minimum and maximum intra-ocular distance when selecting eye candidate pairs.
- *setSymmetryParam(value)* - sets the symmetry degree for the fast radial symmetry transform method.

FaScore class

The *FaScore* class is the one in charge of scoring both the faces and pair of eye centres, and it does so by making use of the *MultivariateNormal* class, which it instantiates into two objects for the faces multivariate normal distribution and the eyes one. The parameters used for the two distributions are the ones previously determined experimentally in Section 3.6 for the eyes, and in Section 3.7 for the eyes pair centres.

scoreDetectionProbabilities(face)

This method evaluates the probabilities of both the face and eyes pair centres for the attributes of the face passed by argument by calling the *scoreFaceDetection* and *scoreEyesDetection* methods. The results are stored inside of the face structure.

scoreFaceDetection(face, eyeLeft, eyeRight)

This method evaluates the probability of the face by taking into account the configuration conformed by the face and eyes bounding boxes. In order to do so, it measures the parameters previously selected in Section 3.6 and calculates the inverse cumulative probability of them belonging to the multivariate normal distribution.

scoreEyesDetection(eyeLeft, eyeRight)

This method evaluates the probability of the eyes pair centres by measuring the parameters previously selected in 3.7 and calculating the inverse cumulative probability of them belonging to the bivariate normal distribution.

RadialSymmetry class

This class implements the *fast radial symmetry transform* algorithm, as well as some smaller algorithms that are required for it to work. The class is structured such that each instantiated object corresponds to the processing of a specific image, being able to perform multiple executions for different values of the transform parameters over the same image. By not allowing the processing of different images inside the same transform, the common steps of the algorithm can be recycled, only calculating them once at the object constructor.

RadialSymmetry(image, width, height)

This method corresponds to the class constructor, and is in charge of storing the image to be processed and generating its vertical and horizontal Laplacian convolution, as well as the map of gradient intensities.

~RadialSymmetry(image, width, height)

this method corresponds to the class destructor, and is in charge of freeing the memory for the map of gradient intensities and vertical and horizontal Laplacian convolutions, which were allocated by the constructor.

applyTransform(rad, alpha, beta, ret)

This method implements the *fast radial symmetry transform*. When called, it applies the transform over the image assigned to the object by using the symmetry degree value *alpha* passed by parameter. The result is stored to the buffer of the same size of the original image *ret*, which is passed by parameter.

convolute2d(image, width, height, direction, ret)

This method applies a transform over the provided image by applying a squared 5-by-5 matrix that performs a Laplacian transform over the direction specified by the parameter *direction*, and performs a Gaussian smoothing in the orthogonal direction. The resulting transformed image is stored in the *ret* buffer, which is passed by parameter and is of the same size of the image.

normalizeMatrix(matrix, nPos, ret)

This method performs a normalization over the input matrix *matrix* which *nPos* positions inside the array. The normalized matrix is stored inside the *ret* buffer of the same size of the original matrix.

MultivariateNormal class

This class implements a multivariate normal distribution, allowing for the generation of random samples following the probability distribution defined by the multivariate normal, as well as to calculate the Probability Density Function (PDF) and Cumulative Distribution Function (CDF) for a given instance.

MultivariateNormal(nDims, means, covariances)

This method is the constructor for the class, and its main task is to assign the multivariate distribution means and covariances, as well as to draw an initial number of random samples from the distribution and to calculate their probability density. These random samples are used for calculating the probability density of a sample when required. The higher the number of generated samples is, the better the cumulative density estimation, with a minimum of 31 samples required to approximate the distribution. In this implementation, 1000 samples are initially generated.

generateSample(): sample

This method generates and returns a random sample following the probability distribution defined by the multivariate normal assigned to this class.

getProbabilityDensity(sample): probability

This method returns the probability density of the sample passed by parameter.

getCumulativeDistribution(sample): cumulative

This method returns the cumulative distribution of the sample passed by parameter. In order to calculate it, the probability density of the sample is obtained, and the proportion of the initially generated samples having a higher or equal probability density is calculated. This proportion is returned as the cumulative distribution for that value.

FaCascadeClassifier class

This class is a class specialization that inherits the *CascadeClassifier* class provided by the *OpenCV* library. It is implemented in order to allow the class to be constructed by passing it directly a string with the contents of the XML classifier, which allows for the storage of the classifiers data inside of the library source code instead of depending on an external file. Thus, the only method overwritten by the *FaCascadeClassifier* class is the constructor.

4.3 Sequence diagrams

In this the interaction between classes and methods is explained for the different use cases, as well as for the library initialization. In order to do so, sequence diagrams are used to visualize these interactions, also describing the general flow of the use case.

4.3.1 Library initialization

The library initialization is the process executed when the library is first loaded by the application. This initialization consists on setting up the classifiers required in order to detect the face and eyes bounding boxes, as well as the multivariate normal distributions used to estimate the detection probabilities.

The sequence diagram shown in Figure 19 describes the execution flow of the library when initialized. As it is shown, the main task of the initialization process is to create the object describing the library configuration, to create both the face and eye cascade classifiers and to create the *FaScore* instance in charge of estimating the face and eye centres probabilities.

The *FaScore* instance, when initialized, is in charge of creating both the face and eye centres multivariate distributions used to estimate the detection probabilities. These distributions are initialized by using the means vectors and covariances matrices previously determined in Section 3.6 for the faces and in Section 3.7 for the eye centres. The *MultivariateNormal* instances, in turn, generate a set of instances for the defined distributions and calculate their probability density when initialized.

The *FaConfiguration* instance, which is assigned to the *Faceanalyser* object representing the library, is in turn in charge of setting the initial values for the methodology parameters, which were found experimentally over a dataset of 1000 images.

Finally, the face and eye detectors initialization is performed by loading the XML strings representing the cascade of AdaBoost classifiers, which are embedded into the library code, into their corresponding *FaCascadeClassifier* instances.

4.3.2 Parameters definition

The *parameters definition* use case is in charge of modifying the values for the methodology parameters, stored inside of the *FaConfiguration* class. This class implements a series of methods to modify the parameter values when required, which are called from the *Faceanalyser* class when setting the methodology parameters. This process is represented in the sequence diagram shown in Figure 20.

The *FaFace* instance processing the image, on the other hand, queries this class when it requires the value of one of these parameters.

4.3.3 Faces and eye centres detection

The *faces and eye centres detection* use case is in charge of performing the faces detection and eye centres location tasks, which are the central objectives of the library. The use case takes an image as input and analyses it, returning an output structure indicating the number of faces detected, and for each detection, the data on the face and eyes bounding boxes and eye centres, as well as their corresponding probabilities.

The sequence diagram shown on Figure 21 shows the execution flow for this use case. The first step of the process is the creation of a worker object of the *FaFace* class, which is in charge

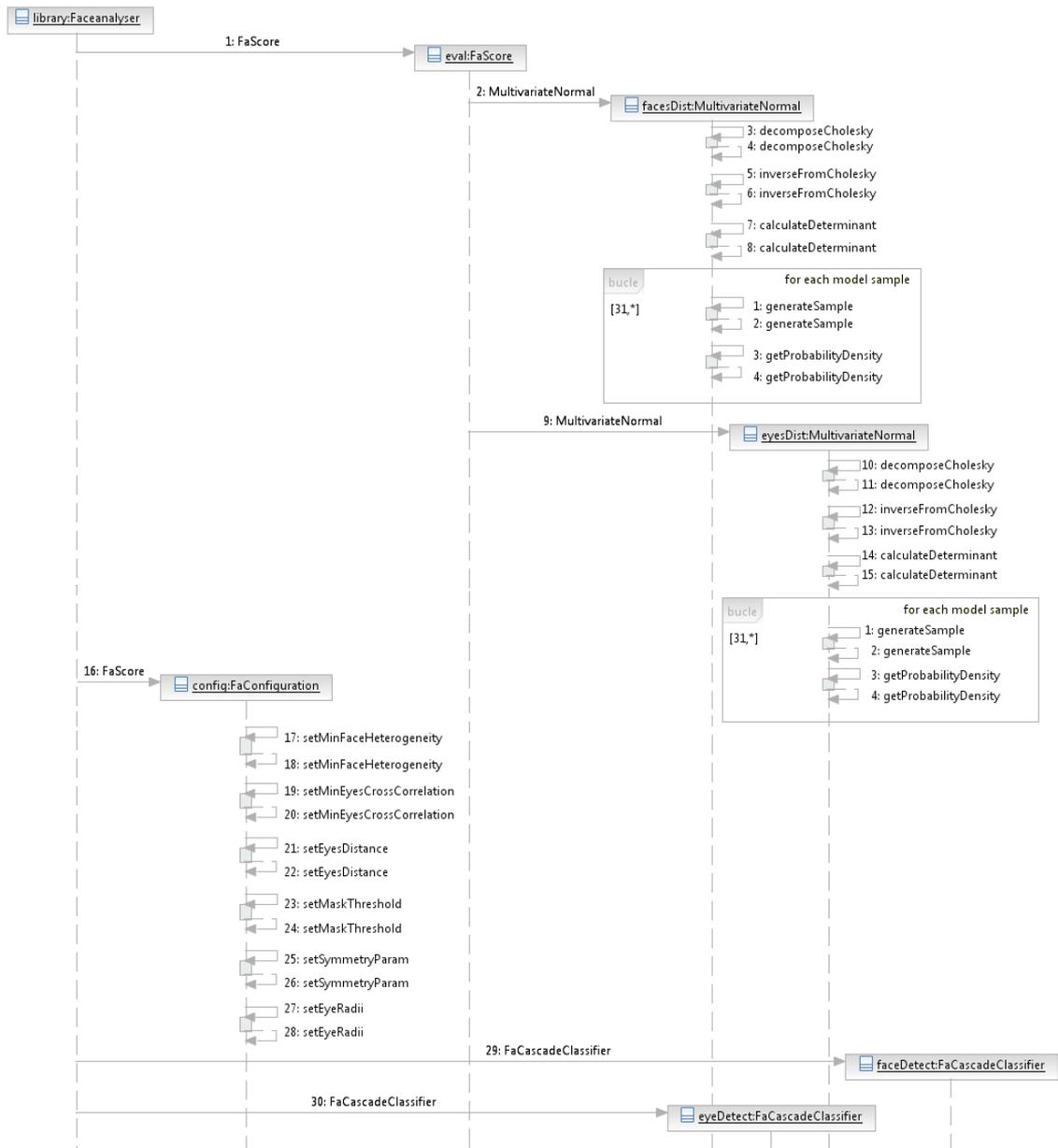


Figure 19: Sequence diagram describing the library initialization process.

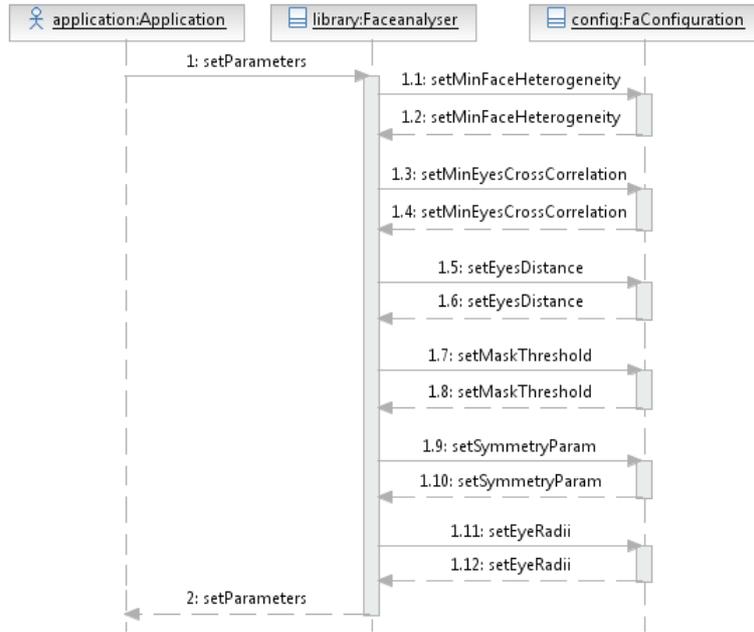


Figure 20: Sequence diagram describing the *parameters definition* use case.

of performing the detection procedure itself upon initialization. This object first performs face detection on the image passed through the constructor, which yields a set of candidates faces over each one of which the eyes are detected. Afterwards, the best eyes pair for each face is determined by selecting the one maximizing the face probability, which is measured by using the *FaScore* instance created during the library initialization.

If a valid eyes pair is found for a given face, the *FaFace* constructor creates an instance of the *SymmetryTransform* class for each eye, which generates the gradient transforms and gradient intensity maps for the given eye image. Afterwards a series of fast radial symmetry transforms are executed using this class in order to determine the optimal eye radius for the eye, and the eye centre is determined from the optimal eye radius symmetry map merged with the neighbouring symmetry maps. The *FaFace* objects are destroyed after determining the eye centres.

With the eye centres located for the accepted face detections, the face and eye centres probabilities are estimated by using the *FaScore* class, which is in charge of measuring the features for both the face configuration and eye centres, afterwards passing them to their corresponding *MultivariateNormal* instances in order to calculate the inverse cumulative densities for these instances.

Once the object initialization finishes, the results can be gathered by calling the two getter methods of the *FaFace* class, which return the number of faces detected and the vector containing the information on these faces.

4.3.4 Results elimination

The *results elimination* use case is used to free the previously allocated memory region containing the information on the detected faces, which was obtained through the *faces and eye centres detection* use case.

The sequence diagram shown on Figure 22 describes the implementation of the use case, which is simple and contained inside the *Faceanalyser* instance representing the library. The process consists on iterating over the list of face detections contained inside of the results structure and

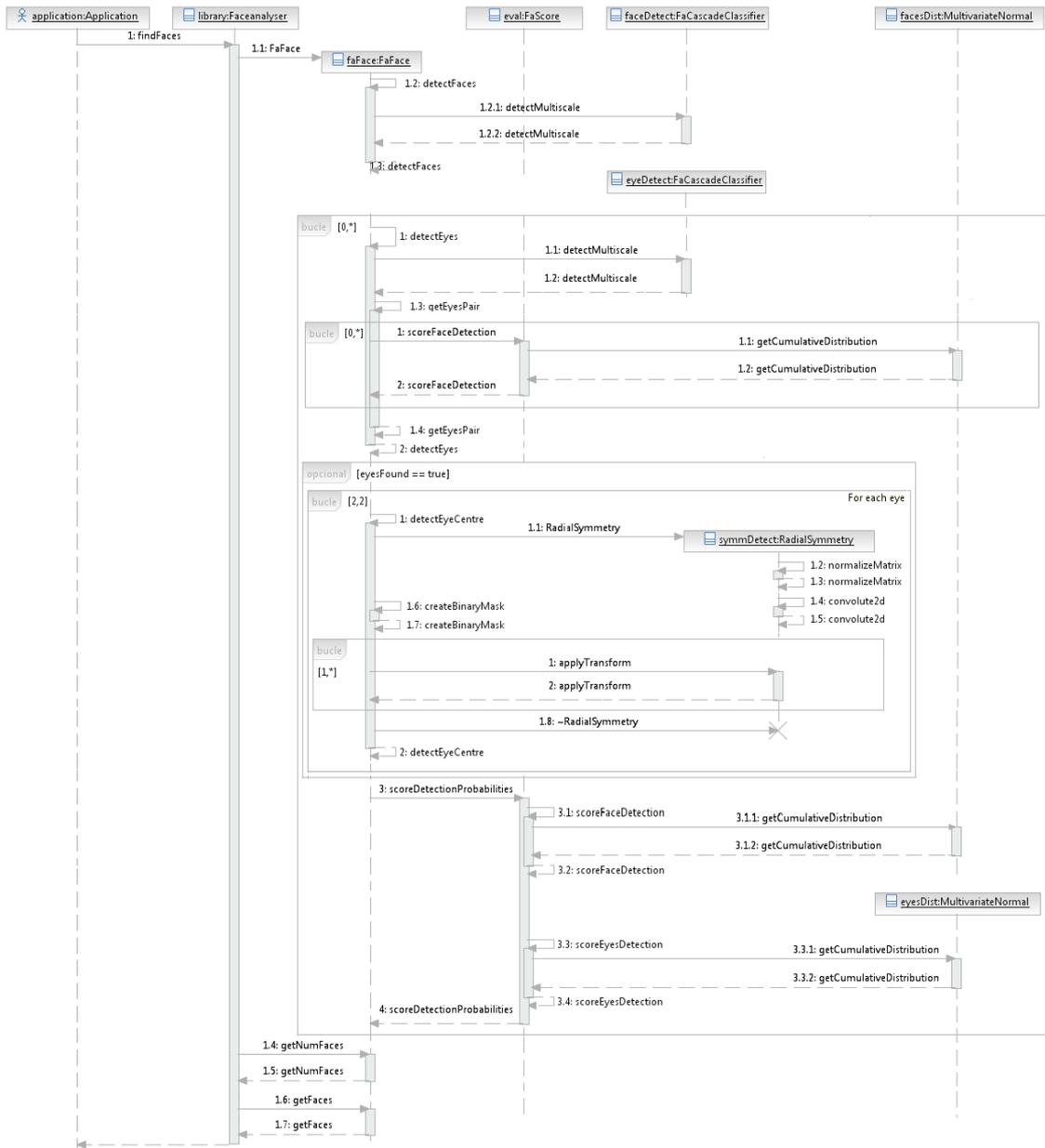


Figure 21: Sequence diagram describing the *face and eye centres detection* use case.

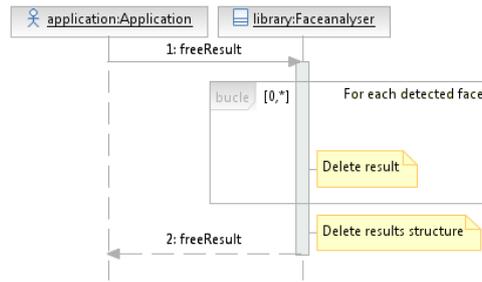


Figure 22: Sequence diagram describing the *results elimination* use case.



Figure 23: Sequence diagram describing the *error message obtention* use case.

freeing the memory regions for these detections individually. Once all of the face structures are freed, the general results structure is also deleted.

4.3.5 Error message obtention

The *error message obtention* use case is a simple use case which uses the error code obtained from the last processed image to select a pre-defined error string. A reference to this error string is returned by the use case. The sequence diagram for this use case is shown in Figure 23.

5 Experimental methodology

The algorithm is evaluated both in terms of the recognition rate (accuracy) and the temporal cost evaluated in function of different parameters. In order to assess the accuracy and temporal cost of the proposed algorithm both for face detection and eye centre location, the 10-fold cross-validation technique is used over a dataset of 1000 images, selecting the parameters for the method over the training partitions, and then measuring both accuracies over the test fold.

5.1 The tested dataset

The dataset used to perform the tests consists on a set of 1000 images, which is balanced both in terms of gender and ethnicity. More precisely, the dataset consists on 200 images for each one of five major ethnicity groups, with 100 images for each gender and ethnicity. The ethnic groups taken into account are the following ones:

- **European** - This group includes the population of Europe, the northern-most part of Africa, and most of the north American population.
- **Middle eastern** - This group includes both the population of middle east and part of the north-east African population.
- **Asian** - This group includes the various ethnic groups of Asia and the islands connecting Asia with Australia.



Figure 24: Sample images of the dataset showing the variability of the images in terms of ethnicity and gender.

- **African** - This group includes the population of the central and southern Africa.
- **South American** - It includes the various ethnicities native to central and southern America.

The goal of using a balanced dataset is to prevent the algorithm from being evaluated over a non-representative sample of the population. Also, by using a dataset with a big number of individuals, it is possible to approximate the face detection accuracy binomial distribution, which measures the face detection ratio, to a normal distribution [17]. The variability of the dataset is shown qualitatively in Figure 24, where sample images are shown for each ethnicity and gender.

In order to assess the correctness of the results obtained by the algorithm, a ground truth for the data set is required against which the results can be compared. This ground truth is created by manually tagging the features to be detected into the images, that is, manually marking the upper-left and bottom-right corners of the face bounding box and the eye centres for each image.

5.2 Accuracy analysis

The accuracy of the algorithm is measured by using a 10-fold cross-validation procedure, which allows for the use of the whole dataset both for parameter selection and accuracy measurement without over-fitting the model.

In order to perform the 10-fold cross-validation, first the dataset is divided into 10 parts, each one of them with equal number of individuals for each ethnicity and gender. With the dataset divided into 10 parts, a parameter selection and accuracy validation is performed, where 9 out of the 10 parts are used for selecting the parameter, and the remaining part is used to validate the accuracy of the model with the parameters selected at the previous step. This is done 10 times, at each step using a different part for testing. This procedure is better illustrated in Figure 25.

During the parameter selection step of each fold, the different parameters affecting the algorithm performance are adjusted, either individually or pair-wise with another parameter, when both parameters are inter-related. All of the parameters affect either the false positives discarding

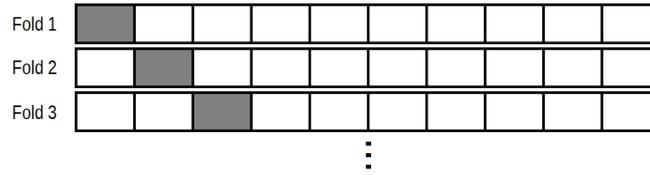


Figure 25: Partitions selection for each 10-fold cross-validation fold. At each fold, a different partition is selected as test partition (coloured grey), using the rest for parameter selection (coloured white).

procedure for the face and eye bounding boxes, or the eye centre detection accuracy.

For the parameters affecting the false positives discarding, the criterion for selecting a value for a given parameter is to select the parameter value that maximizes the number of true positives, while at the same time but less importantly minimizes the number of false positives. The rest of parameters not being optimized are left unused, with no false positive being filtered by them. By selecting the parameter values in this manner, the overall algorithm false positives are reduced while keeping all of the true positives. The parameters for filtering false positives are the following ones:

- **Minimum face heterogeneity** - Minimum intensity heterogeneity required in order to consider a face bounding box as valid.
- **Maximum eyes pair cross-correlation** - Maximum allowed cross-correlation between the pixel intensities of two eye bounding boxes when extrapolating one of them from the other.
- **Allowed intra-ocular distance** - Includes both the minimum and maximum allowed distance between a pair of eyes bounding box centres. Both parameters are selected simultaneously through a grid search.

For the parameters affecting the eye centre, the criterion for selecting a given parameter value is to maximize the area under the curve for the error margin versus accuracy plot. This plot represents in one dimension the allowed euclidean distance between the detected eye centre and the ground truth, divided by the face width, and in the other dimension the number of true positives given the allowed error margin. If a higher percentage of the area falls under the curve, it means that the overall precision of the method is higher. The parameters affecting the eye centre detection accuracy are the following ones:

- **Radial symmetry test radii** - Includes both the minimum and maximum test radius values to be tested when determining the optimal radius of the iris. These two parameters must be optimized at the same time through a grid search.
- **Eyes binary mask threshold** - This threshold defines the intensity value for the normalized eye bounding box pixels under which the eye centre is to be searched. Any eye bounding box pixel with a higher value is ignored.
- **Fast radial symmetry transform degree** - This parameter determines the degree of symmetry that is forced when applying the fast radial symmetry transform algorithm. A higher value requires a higher level of symmetry at the given region in order to detect it.

Once the 10-fold cross-validation has determined the optimal parameters for each fold and obtained the results of processing the images of their corresponding test folds, the results for all the test folds are put together, resulting in a set of processed test images containing all of the images of the original dataset. Over this dataset of processed images, the accuracy is measured

for both the face detection accuracy and eyes centre detection accuracy, also analysing other characteristics of the model through the ROC curve and other criteria.

Finally, the algorithm accuracy is compared with the commercial methodology developed by *Neurotechnologies* over a different dataset of 266 instances for which the detection results using the commercial methodology are available.

5.3 Temporal cost analysis

In order to evaluate the temporal cost of the methodology, the same dataset used for the accuracy assessment is used, with the parameters fixed to the optimal values found when taking into account the whole dataset for parameter selection, except for the parameters being evaluated. The experimental set-up to evaluate the temporal cost of processing an image consists on an *cc2.8xlarge* Amazon instance with a *Windows Server 2012* operating system installed. This configuration consists on two *Intel(R) Xeon(R) CPU E5-2670* CPUs at *2.60GHz*, each one consisting on 8 cores with hyper-threading, which is effectively equivalent to a 32-core machine at *2.60GHz*. It also has *60GB* of DDR3 memory. Each image is processed 31 times in order to obtain a statistically significant mean processing time, with the whole workload being divided into 14 independent processes run simultaneously.

The temporal cost is first evaluated in function of the *image surface area*, which determines the search space for the faces, and the *minimum face heterogeneity* parameter, which determines the search space for the eyes. These two characteristics, while one being dependant on the data and the other dependant on the methodology, both directly affect the search spaces for the cascade classifiers. Afterwards, the temporal cost is evaluated relative to the number of test radii for the eyes centre location, since it has a direct impact on the number of arithmetic operations to perform by determining the number of testing symmetry transforms.

6 Experiments

6.1 Parameter selection

In the following subsections, each parameter or set of inter-related parameters is analysed for each individual fold, selecting its optimal value. In order to do so, either a linear search or a grid search is performed over a set of test values of the parameters, afterwards selecting the parameter value that maximizes the rate of true positives. As a second selection criteria, when optimizing the False Positives filtering parameters, the value minimizing the total number of False Positives is selected.

6.1.1 Minimum face heterogeneity

The *minimum face heterogeneity* parameter describes the minimum heterogeneity allowed for the intensity values of a face candidate bounding box pixels in order to prevent it from being discarded. This parameter affects the false positive discarding process, and as such must be optimized such that the number of false positives is as low as possible without discarding any true positive.

The results from linearly testing a range of values for the parameter are seen in Figure 26, where it is shown that for all 10 folds the optimal value for the parameter is found at 0.065. Even though this parameter has only a statistically significant improvement for values of the parameter over 0.1, it is the lowest value with the best true positive rate.

Appendix A shows the precise True Positive values obtained with a 95% confidence interval,

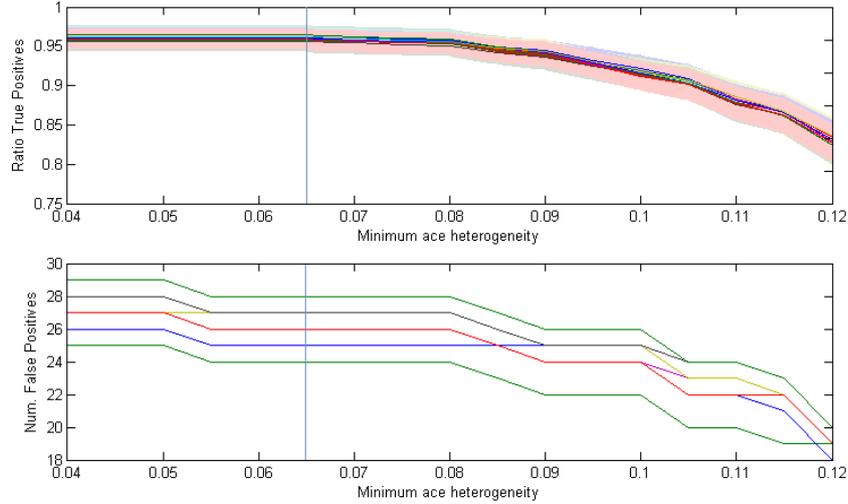


Figure 26: Plots representing the True Positives rate and total number of False Positives relative to the selected value for the minimum face heterogeneity parameter.

	Value	TP rate	FP
Fold 1	0.065	0.9611 ± 0.9611	26
Fold 2	0.065	0.9644 ± 0.0062	24
Fold 3	0.065	0.9589 ± 0.0066	26
Fold 4	0.065	0.9622 ± 0.0064	26
Fold 5	0.065	0.9589 ± 0.0066	26
Fold 6	0.065	0.9600 ± 0.0065	27
Fold 7	0.065	0.9567 ± 0.0068	27
Fold 8	0.065	0.9600 ± 0.0065	25
Fold 9	0.065	0.9578 ± 0.0067	28
Fold 10	0.065	0.9589 ± 0.0066	26

Table 5: Table showing the selected value for the *minimum face heterogeneity* parameter at each fold, with the obtained True Positive rate, its 95% confidence interval and total number of False Positives for this value.

with the optimal value for each fold highlighted. From this table, the optimal value for the parameter at each fold, as well as its accuracy and confidence interval can be obtained. These values are the ones shown at Table 5.

6.1.2 Maximum eyes pair cross-correlation

The *maximum eyes pair cross-correlation* parameter describes a maximum value threshold for the eyes bounding box cross-correlation of the pixel intensities when one of the eye bounding boxes is extrapolated from the other. If the cross-correlation value is below the parameter value, the extrapolated eye bounding box is accepted as valid, allowing for the face to also be considered valid. A too high value for the parameter increases the False Positives rate by detecting invalid eyes, while a value that is too low decreases it by preventing the detection of some eye candidates.

In order to determine the optimal values for the *maximum eyes pair cross-correlation* parameter, a range of values has been tested in a linear search at each fold. The results of these linear searches can be seen in Figure 27, where it is shown that in most cases the highest True Positives value minimizing the number of False Positives is of 0.17, although the confidence intervals indicate that at the 95% Confidence Interval there is no significant difference in the True Positive

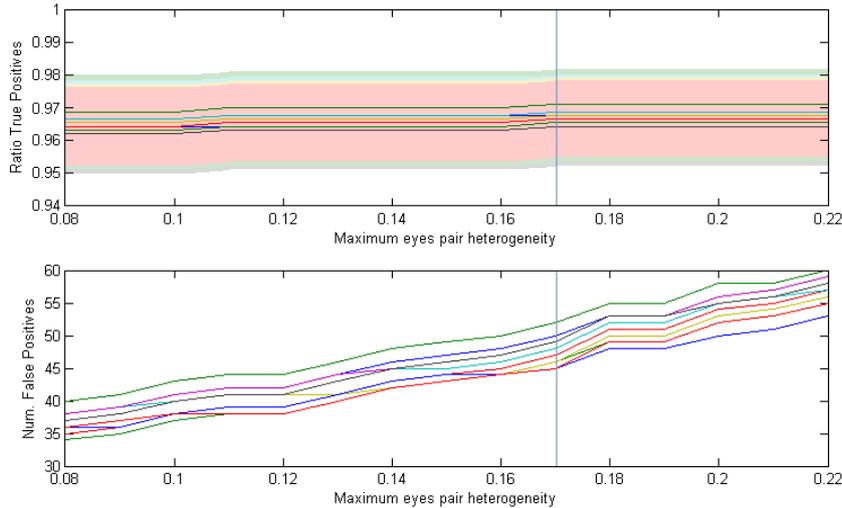


Figure 27: Plots representing the True Positives rate and total number of False Positives relative to the selected value for the maximum eyes pair cross-correlation parameter.

	Value	TP rate	FP
Fold 1	0.110	0.9678 ± 0.0059	42
Fold 2	0.170	0.9711 ± 0.0056	46
Fold 3	0.170	0.9656 ± 0.0061	47
Fold 4	0.170	0.9689 ± 0.0058	48
Fold 5	0.170	0.9667 ± 0.0060	49
Fold 6	0.170	0.9678 ± 0.0059	46
Fold 7	0.170	0.9644 ± 0.0062	49
Fold 8	0.170	0.9656 ± 0.0061	45
Fold 9	0.170	0.9656 ± 0.0061	52
Fold 10	0.170	0.9667 ± 0.0060	45

Table 6: Table showing the selected value for the *maximum eyes pair cross-correlation* parameter at each fold, with the obtained True Positive rate, its 95% confidence interval and total number of False Positives for this value.

values for the explored range.

Appendix B shows the precise True Positive rate obtained with a 95% confidence interval, for each parameter value at each fold. From this table, the optimal value for the parameter at each fold, as well as its accuracy and confidence interval can be obtained. These values are the ones shown at Table 6.

6.1.3 Allowed distance between eyes

The *allowed distance between eyes* is a set of two parameters defining the minimum and maximum distances allowed between the eyes bounding box centres. Since both parameters play a role in the process of discarding eye pairs candidates, and in many cases there are multiple pairs to select even if some valid ones are discarded, these parameters must be optimized at the same time.

A grid search is used in order to optimize both parameters, where one dimension of the search corresponds to the minimum distance, and the other to the maximum. Figure 28 shows the results of the grid search over the 10 folds. Each pair of grid plots represents the True Positive rates and number of False Positives for a given fold, with the colour representing the TP/FP value

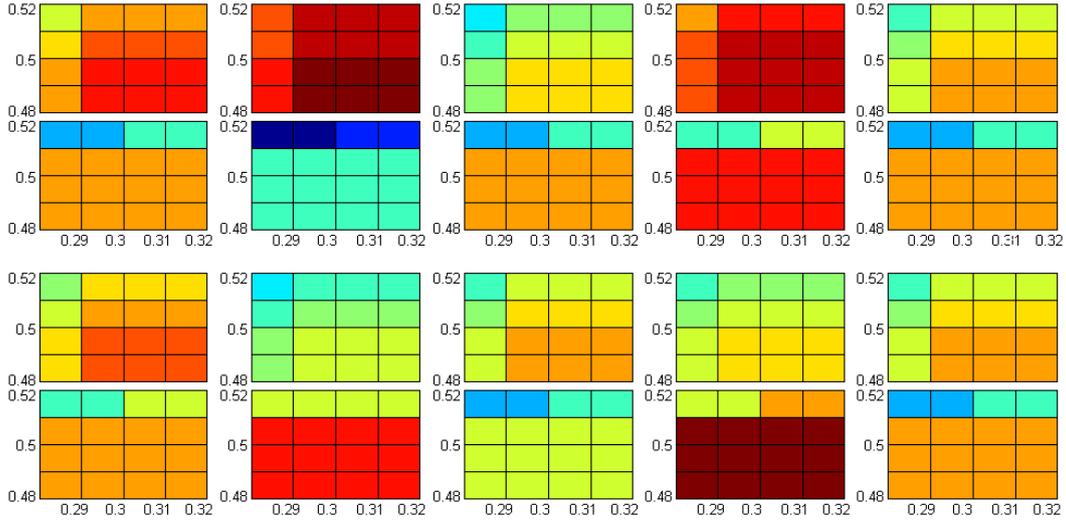


Figure 28: True Positives rate and total number of False Positives surfaces for the minimum and maximum distance between eyes, for each fold.

	Min. Value	Max. Value	TP rate	FP
Fold 1	0.29	0.50	0.9656 ± 0.0061	26
Fold 2	0.29	0.50	0.9678 ± 0.0059	24
Fold 3	0.29	0.50	0.9622 ± 0.0064	26
Fold 4	0.29	0.51	0.9667 ± 0.0060	24
Fold 5	0.29	0.50	0.9633 ± 0.0063	26
Fold 6	0.29	0.50	0.9644 ± 0.0062	26
Fold 7	0.29	0.50	0.9611 ± 0.0064	27
Fold 8	0.29	0.50	0.9633 ± 0.0063	25
Fold 9	0.29	0.50	0.9622 ± 0.0064	28
Fold 10	0.29	0.50	0.9633 ± 0.0063	26

Table 7: Table showing the selected values for the *intra-ocular distance* minimum and maximum values at each fold, with the obtained True Positive rate, its 95% confidence interval and total number of False Positives for these values.

at each point of the grid. It can be seen that the optimal True Positives rate value minimizing the number of False Positives is found in most cases for a minimum intra-ocular distance of 0.29 and a maximum intra-ocular distance of 0.5.

Appendix C shows the precise True Positive rate obtained with a 95% confidence interval, for each set of parameter values at each fold. From these tables, the optimal values for the parameters at each fold, as well as its accuracies and confidence intervals can be obtained. These values are the ones shown at Table 7.

6.1.4 Radial symmetry test radii

The *radial symmetry test radii* parameters are a set of two parameters defining the lower and upper bound for the test radii to take into account when looking for the optimal radius of the iris to use when detecting the eye centre. The selection of a good range of test radii is critical for locating the eyes centre, since testing values that are too small or too big could mean that symmetries other than that of the iris could be detected, locating the eye centre in the wrong place.

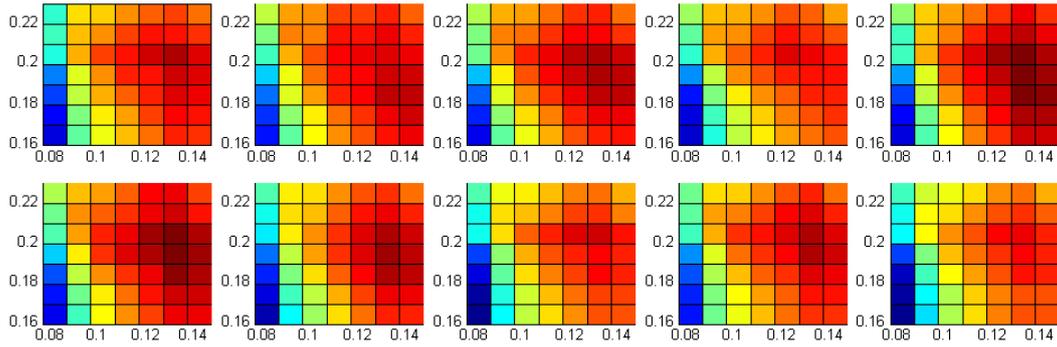


Figure 29: True Positives rate surfaces for the minimum and maximum test radii over the radial symmetry method, for each fold.

	Min. Radius	Max. Radius	TP rate
Fold 1	12	21	0.97628 ± 0.00079
Fold 2	10	21	0.97624 ± 0.00080
Fold 3	12	21	0.97630 ± 0.00080
Fold 4	12	20	0.97605 ± 0.00080
Fold 5	11	21	0.97651 ± 0.00079
Fold 6	12	21	0.97652 ± 0.00080
Fold 7	11	21	0.97632 ± 0.00080
Fold 8	12	21	0.97614 ± 0.00081
Fold 9	12	21	0.97629 ± 0.00081
Fold 10	12	21	0.97600 ± 0.00082

Table 8: Table showing the selected values for the *radial symmetry test radii* minimum and maximum test radius at each fold, with the obtained True Positive rate and its 95% confidence interval for these values.

In order to determine the optimal minimum and maximum test radii, a grid search is performed testing the accuracy for both parameters at the same time. These parameters must be optimized jointly, since the detection of high symmetry values for small radius values could discard other symmetry values detected at the higher end of the test radius values, and the same could happen in the opposite direction. Figure 29 shows the grid search results by using a True Positive rates surface for each fold, showing that the True Positives rate increases the closer both values are to a minimum test radius between 0.13 and 0.14, and a maximum test radius between 0.20 and 0.21. This indicates that a wider range of test radii is not necessarily better, existing an optimal exploration range.

The precise True Positive values within a 95% Confidence Interval are shown in Appendix D for each fold and pair of tested parameter values. These results are summarized in Table 8, where the best parameter values for the minimum/maximum test radius are shown, along with its True Positives rate within a 95% Confidence Interval.

6.1.5 Eyes binary mask threshold

The *binary mask threshold* defines the intensity value of the eye bounding box with normalized intensities for which all intensity values of the bounding box under this value are marked as positive for the binary mask. This parameter influences the region of the eye bounding box to be considered when looking for the eye centre.

In order to determine the optimal value for the parameter, a linear search is performed for a range of value candidates. The results from this search are shown in Figure 30. From this figure,

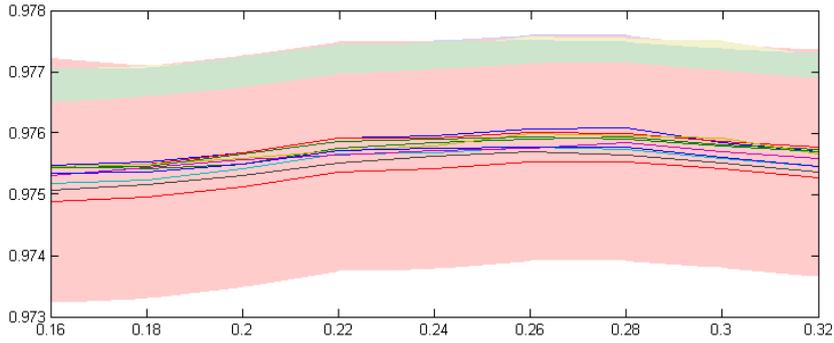


Figure 30: Area under the curve for the error margin versus True Positives rate plot with a 95% confidence interval, relative to the eyes binary mask threshold, for each fold.

	Value	TP rate
Fold 1	0.260	0.9761 ± 0.0008
Fold 2	0.260	0.9759 ± 0.0008
Fold 3	0.260	0.9760 ± 0.0008
Fold 4	0.260	0.9758 ± 0.0008
Fold 5	0.280	0.9758 ± 0.0008
Fold 6	0.260	0.9760 ± 0.0008
Fold 7	0.260	0.9757 ± 0.0008
Fold 8	0.260	0.9758 ± 0.0008
Fold 9	0.260	0.9759 ± 0.0008
Fold 10	0.260	0.9755 ± 0.0008

Table 9: Table showing the selected value for the *eyes binary mask threshold* parameter at each fold, with the obtained True Positive rate and its 95% confidence interval for this value.

it can be seen that the optimal *eyes binary mask threshold* is found for a value between 0.24 and 0.28 depending on the fold, although the optimal value improvement is not significant within the 95% confidence interval.

The precise values for each fold, as well as its confidence interval, are shown on Appendix E, where the table shows the accuracy within a 95% confidence interval for each parameter test value and fold, highlighting the best accuracy value for each fold. The selected values for each fold, along with its True Positives rate within a 95% confidence interval and the number of False Positives are shown in Table 9.

6.1.6 Fast radial symmetry degree

The *fast radial symmetry degree* parameter defines the degree of symmetry required in order to consider that there is a radial symmetry at a given point and for a given radius. The higher this parameter is, more circular must the analysed component be in order to be detected when applying the fast radial symmetry transform method. By increasing this parameter value, it is possible to discard some symmetries not corresponding to the retina, since they have a lower symmetry degree. A parameter value that is too high might discard the retina altogether.

In order to select a value for this parameter, a linear search is performed for each fold over a range of test values. The result of the linear search can be seen in Figure 31, where it is shown that for all folds the optimal value of the parameter can be any value between 1 and 1.75, although the True Positives rate is significantly better for a 95% confidence interval only compared to values of the parameter higher or equal than 3.

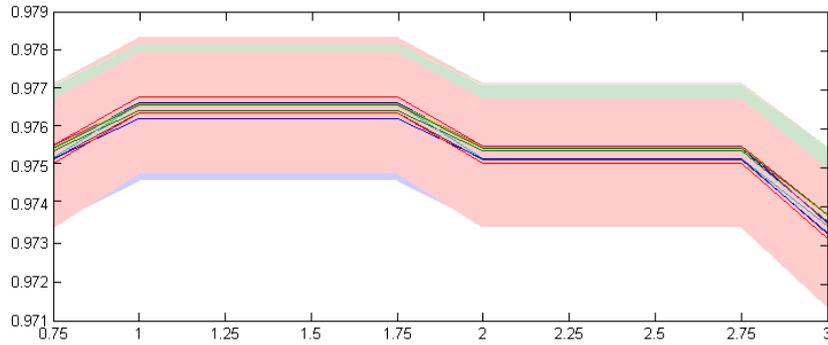


Figure 31: Area under the curve for the error margin vs. True Positives rate plot with a 95% confidence interval, relative to the *fast radial symmetry degree* parameter, for each fold.

	Value	TP rate
Fold 1	1.5	0.9766 ± 0.0008
Fold 2	1.5	0.9764 ± 0.0008
Fold 3	1.5	0.9768 ± 0.0008
Fold 4	1.5	0.9766 ± 0.0008
Fold 5	1.5	0.9766 ± 0.0008
Fold 6	1.5	0.9766 ± 0.0008
Fold 7	1.5	0.9764 ± 0.0008
Fold 8	1.5	0.9762 ± 0.0008
Fold 9	1.5	0.9766 ± 0.0008
Fold 10	1.5	0.9764 ± 0.0008

Table 10: Table showing the selected value for the *fast radial symmetry degree* parameter at each fold, with the obtained True Positive rate and its 95% confidence interval for this value.

The specific values for the True Positives rate within a 95% confidence interval for each parameter value and fold can be seen in Appendix F. The selected parameter values at each fold, along with its True Positives rate value within a 95% confidence interval are summarized on Table 10

6.1.7 Selected parameter values

In the previous sections, the optimal values for each parameter and fold have been selected. These parameters are used in the following sections in order to obtain the final results for all dataset instances, using the parameters of each fold to evaluate the images on the test partition. Table 11 summarizes the obtained parameter values at each fold.

6.2 Face detection accuracy

This section analyses the obtained accuracies for face detection over the test partitions of each fold, both individually and by taking into account all of the test partitions together. In order to do so, the best parameters found in Section 6.1 for each fold are used to evaluate their corresponding test partitions. Afterwards the results of these evaluations are pooled together in order to use the whole dataset to evaluate the overall method accuracy. The results of this process are shown in Table 12.

It can be seen that while for the different folds the mean accuracy varies in widely and the confidence intervals allow for variations of around a two percent, when taking into account the

	minFH	maxECC	minDE	maxDE	minRSR	maxRSR	thrEBM	radSD
Fold 1	0.065	0.11	0.29	0.50	12	21	0.26	1.5
Fold 2	0.065	0.17	0.29	0.50	10	21	0.26	1.5
Fold 3	0.065	0.17	0.29	0.50	12	21	0.26	1.5
Fold 4	0.065	0.17	0.29	0.51	12	20	0.26	1.5
Fold 5	0.065	0.17	0.29	0.50	11	21	0.28	1.5
Fold 6	0.065	0.17	0.29	0.50	12	21	0.26	1.5
Fold 7	0.065	0.17	0.29	0.50	11	21	0.26	1.5
Fold 8	0.065	0.17	0.29	0.50	12	21	0.26	1.5
Fold 9	0.065	0.17	0.29	0.50	12	21	0.26	1.5
Fold 10	0.065	0.17	0.29	0.50	12	21	0.26	1.5

Table 11: Table showing the selected values for the different parameters of the developed methodology at each fold. The parameters are *minimum face heterogeneity* (minFH), *maximum eyes pair cross-correlation* (maxECC), *min/max allowed distance between eyes* (minDE, maxDE), *min/max radial symmetry test radii* (minRSR, maxRSR), *eyes binary mask threshold* (thrEBM) and *fast radial symmetry degree* (radSD).

	TP rate	FP
Fold 1	0.9500 ± 0.0218	3
Fold 2	0.9100 ± 0.0286	7
Fold 3	0.9700 ± 0.0171	3
Fold 4	0.9400 ± 0.0237	4
Fold 5	0.9700 ± 0.0171	4
Fold 6	0.9600 ± 0.0196	4
Fold 7	0.9900 ± 0.0099	1
Fold 8	0.9700 ± 0.0171	6
Fold 9	0.9800 ± 0.0140	1
Fold 10	0.9700 ± 0.0171	5
	0.9610 ± 0.0061	38

Table 12: Table showing the True Positives rate within a 95% Confidence Interval and number of False Positives over the test partition of each fold and the overall values for the combination of the results at each test partition.

	Men	Women	Both
European	1.0000 ± 0.0000	0.9800 ± 0.0140	0.9900 ± 0.0070
Middle eastern	0.9700 ± 0.0171	0.9400 ± 0.0237	0.9550 ± 0.0147
Asian	0.9900 ± 0.0099	0.9700 ± 0.0171	0.9800 ± 0.0099
African	0.9200 ± 0.0271	0.8800 ± 0.0325	0.9000 ± 0.0212
South American	0.9700 ± 0.0171	0.9900 ± 0.0099	0.9800 ± 0.0099
	0.9700 ± 0.0076	0.9520 ± 0.0096	

Table 13: Test accuracies using all of the test partitions for each ethnicity and gender.

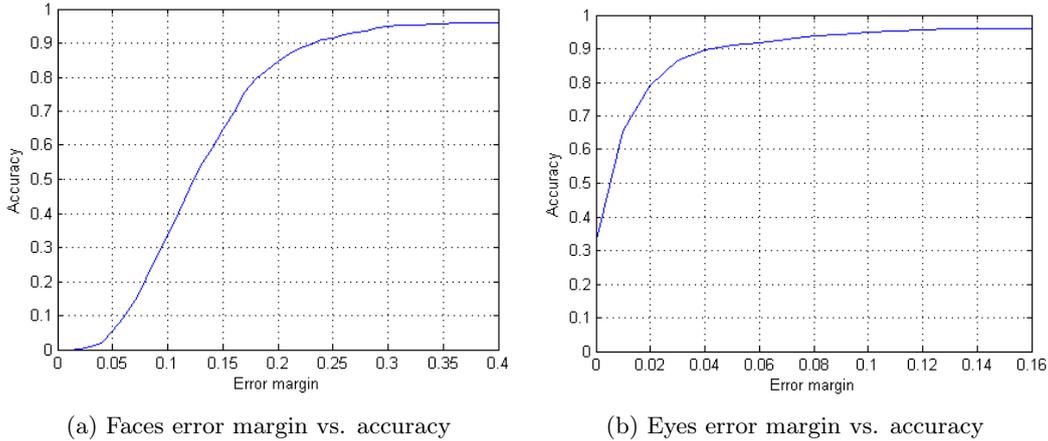


Figure 32: Recall relative to error margin for both face detection (left) and eyes centre location (right).

instances of all test partitions together the algorithm achieves a mean accuracy of 0.9610 ± 0.0061 at the 95% Confidence Interval. Also, over the dataset of 1000 images only 38 False Positives have been found. In Section 6.4 these False Positives are evaluated in more detail by taking into account the probability assigned to them compared to that of the True Positives.

When using the test partitions of all folds to analyse the accuracies at the ethnicity and gender level, as it is shown in Table 13, differences appear. At the gender level, the accuracy of the system is slightly higher in the case of men, with a 97% compared to the 95.2% accuracy in the case of women, difference that is statistically significant. In the case of ethnicities, while the European, Asian and south American detection accuracies are similar, with no statistically significant differences at the 95% Confidence Interval, the middle eastern and African detection accuracies are much lower with a 95% and 90% detection accuracy respectively. These accuracies are much lower specially in the case of women, due to the common use of kerchiefs in the case of middle eastern women, and because of a low colour variability in the face intensities for both African men and women.

6.3 Error margin versus Recall

One of the analysed measures in order to assess the accuracy of the method is the error margin versus recall curve, where the recall (True Positives rate) of the method is evaluated with relation to the allowed error margin. This type of plot applies both to the face detection and eyes centre location, and is useful in order to get a better insight into the accuracy of the methodology when the accuracy itself is not defined by an absolute value, but rather it is defined by a real value. The curves for both face detection and eyes centre detection are shown in Figure 32.

For the face detection, the used measure for the error margin is one minus the proportion of

area overlap between the ground truth face bounding box and the detected one, which is defined as $overlap = 1 - (face_{gt} \cap face_{detected}) / (face_{gt} \cup face_{detected})$.

Figure 32a shows the curve describing the faces detection accuracy relative to the allowed error margin. From that curve, it can be seen that most faces are found with an error margin between the 5% and 20%, and that 90% of the ground truth faces are detected with an area mismatch inferior to the 25%. This amounts to most of the True Positives, since as seen in Section 6.2 the algorithm accuracy is of the 96.1%.

For the eyes centre detection, the used measure for the error margin is the euclidean distance between the ground truth eye centre and the detected one divided by the ground truth intra-ocular distance, that is, $distance = dist(gt_{xe}, detected_{xe}) / gt_w$. By dividing the euclidean distance to the ground truth by the face width, the error margin is made invariant to the scale of the face.

Figure 32b shows the curve describing the eye centres detection accuracy relative to the allowed error margin. In that case, it can be seen that 90% of the ground truth eye centres are located within an error margin of 4% of the face bounding box width. Another feature that can be seen in the face is a second steep increase of the accuracy past the 6% error margin. This second increase corresponds to the detection of other symmetries not corresponding to the eye centre. While the algorithm detects 92% of the eye centres with high precision, in the other 4% of located eye centres (of the 96% of face True Positives) the eye centres are completely missed, being detected somewhere else inside of the eye bounding box.

The algorithm accuracy can be qualitatively evaluated through the examples shown in Figure 33, where processed images for each ethnicity and gender are shown in the two first rows. The last row shows examples of the most common detection errors made by the algorithm.

In the first case, it is shown that some image regions can be considered faces without corresponding to real face regions. This happens because of the region displaying the same intensity symmetries used by the Viola & Jones object detector to locate both the face and the eyes. The second type of error, shown in the second image of the third row, is similar to the first one, but happens specifically because of the detection of a face in the region corresponding to the chin of another face. This happens because of the mouth being interpreted as a pair of eyes in some cases.

In the case of the image on the third row and third column, the face is not detected at all because of the Viola & Jones having problems when the intensity differences between regions being too low. This happens on faces with a low variation on intensity, and is the reason why the accuracy of the overall methodology is lower in the case of the African ethnicity.

The last two error types, corresponding to a bad location of the eye centres, both happen because of the detection of another radial symmetry in a different region of the eye bounding box. In the first case, the error is due to an excessive inclination of the face, which makes the eye estimated eye bounding box for the left eye fall slightly outside of the eye centre. This forces the algorithm to look for other symmetries, in this case taking the shadow of the ocular cavity. In the second case, the algorithm detects the extreme of the eyebrow instead of the real eye centre.

6.4 Precision versus Recall

The precision versus recall curve, also known as the Receiver Operating Characteristic (ROC) curve, is a plot that represents the detected True Positive rate (recall) relative to a given False Positive rate (precision), where the ratios are relative to the total amount of true/false positives detected (not the ground truth). This type of plot displays a linear relation between true/false positive rates when the true positives are homogeneously distributed across the false positives for the analysed selection criteria.

The analysis of the ROC curve can only be applied to the detection of faces, since there are no false positives for the eyes given the true faces, only a certain error margin. Figure 34 shows

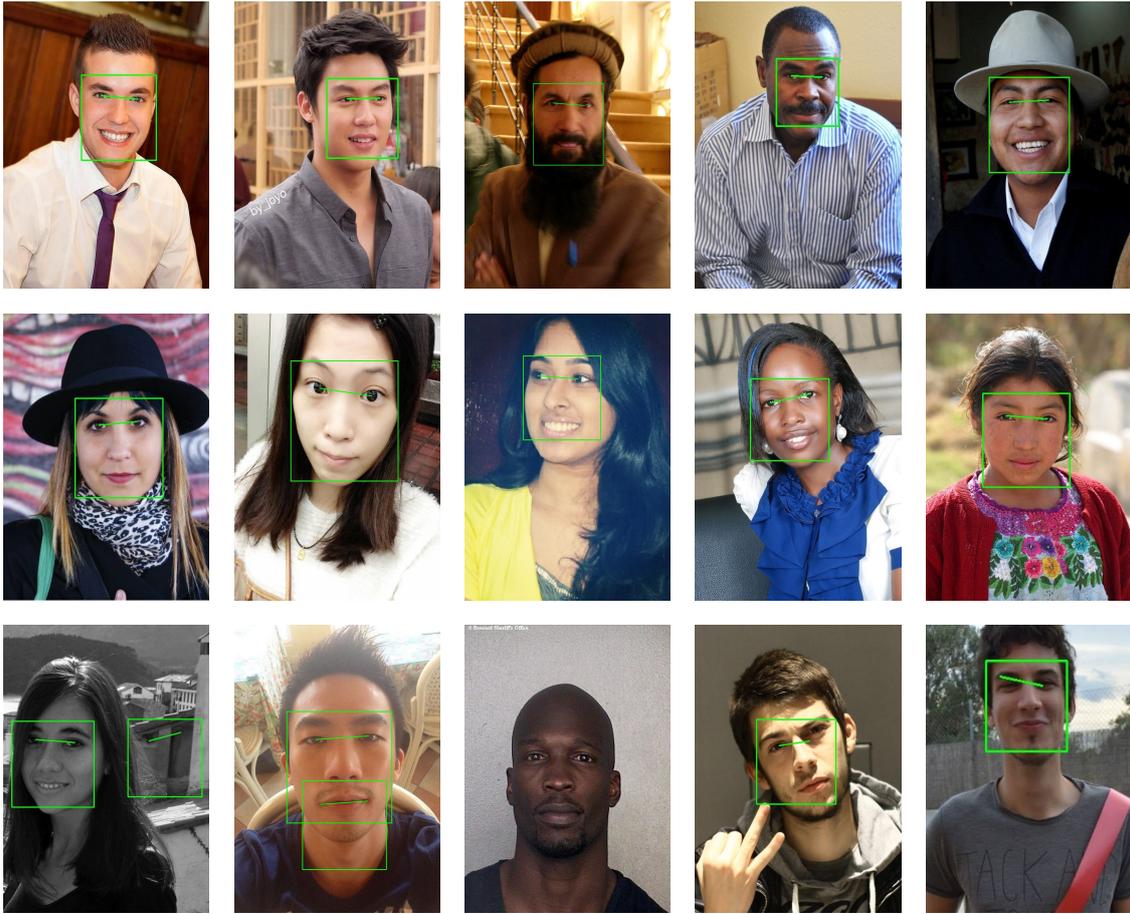


Figure 33: Sample processed images of the dataset showing the variability of the images in terms of ethnicity and gender, as well as the most common detection errors.

the ROC curve for the faces, where the thresholding value used to determine the true/false positive ratios is the face probability, considering all true/false positive faces above a given probability value.

The relation that is found through the ROC curve shows that the relation between True Positives and False Positives is not linear, with many True Positives being found for the False Positives with higher probability of being faces. Up to 90% of True Positives can be found taking into account less than 50% of False Positives, which means that while there is a small number of false positives that are not directly discarded by the methodology, these tend to have a smaller probability assigned, being more easily discarded by the algorithm.

6.5 Comparison with other solutions

In this section a comparison is performed between the developed methodology and the off-the-shelf solution proposed by *Neurotechnologies*. In order to perform this comparison, an independent dataset of 266 images is used, adjusting the parameters of the *Faceanalyser* methodology by using the dataset of 1000 images to perform the same grid search procedure followed in Section 6.1 but without using 10-fold cross-validation. By using an independent dataset for parameter selection and the new dataset for testing both methodologies, over-fitting of the developed methodology is prevented. The optimal parameters found are shown in Table 14.

Figure 35 shows the comparison between both solutions in terms of face detection accuracy and

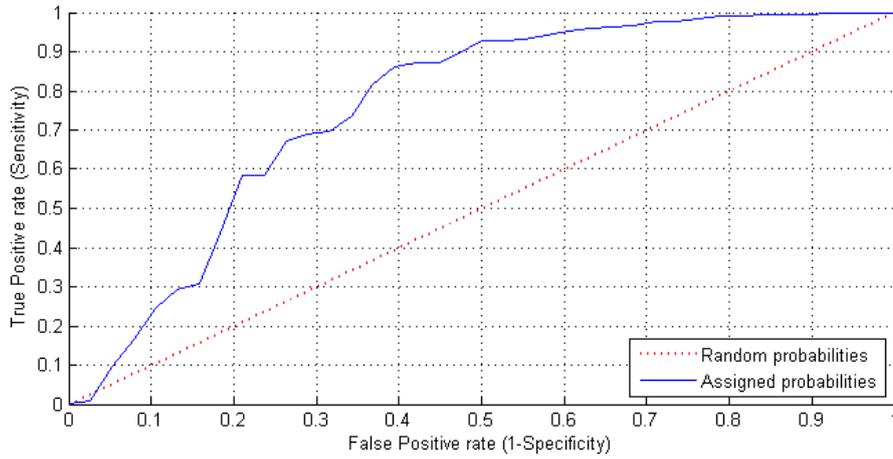


Figure 34: ROC curve for the face detection using the face probability to threshold the parameters.

	Value
Minimum face heterogeneity	0.065
Maximum eyes cross-correlation	0.17
Minimum intra-ocular distance	0.29
Maximum intra-ocular distance	0.50
Minimum eye symmetry test radii	0.12
Maximum eye symmetry test radii	0.21
Eyes binary mask threshold	0.26
Fast radial symmetry degree	1.5

Table 14: Selected parameter values using the full 1000-images dataset in order to compare the *Faceanalyser* and *Neurotechnologies* solutions.

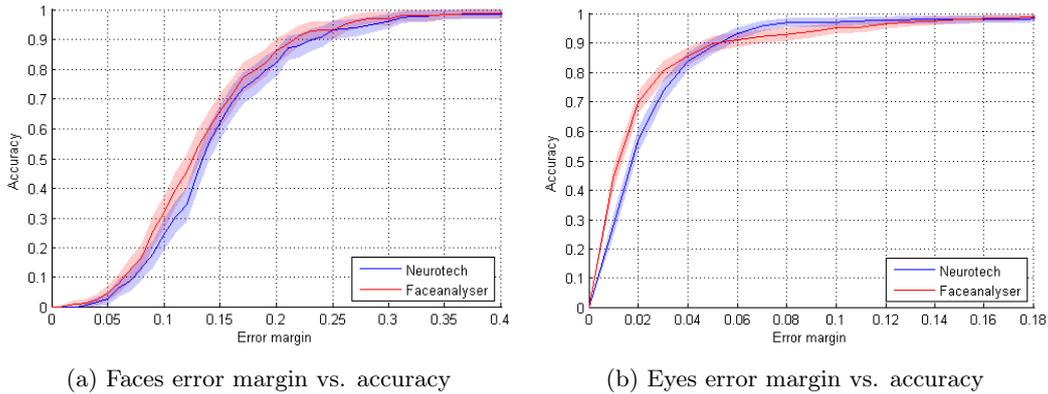


Figure 35: Recall relative to the allowed error margin of both the *Faceanalyser* and *Neurotechnologies* solutions for face detection (left) and eyes centre detection (right) within a 95% Confidence Interval.

eyes detection accuracy. As it can be seen, while the developed method is in average better than the *Neurotechnologies* in both cases, in the case of face detection the confidence intervals slightly overlap for any possible allowed error margin values, which means that at the 95% Confidence Interval it cannot be told which of both methods is better.

In the case of eye centre location, though, there is no overlap of the Confidence Intervals for error margin values lower than the 3% of the face width, which covers about 80% of the processed images in the proposed methodology, and only about the 75% in the case of the *Neurotechnologies* solution. The *Neurotechnologies* solution increases its accuracy relative to the error margin past this point at a higher pace, although not yielding significantly better results. This illustrates a characteristic of the developed methodology that was previously seen in Section 6.3, the detected eye centres are either located with high precision (higher than the commercial solution it is compared against) when correctly detected, or completely missed otherwise because of an alternative symmetry not corresponding to the eye centre being found.

6.6 Temporal cost

The temporal cost for the methodology is evaluated relative to the image size, which is the main constraint of the input data affecting the computational cost, and the algorithm parameters *minimum face heterogeneity* and the *test radii range*, which both affect the amount of computations to be performed on an image. The rest of parameters are fixed to the values found when performing the parameter selection over the whole dataset, which were previously seen in Table 14.

The image size, as it is shown in Figure 36, increases in a linear trend as the surface area of the image increases, with the temporal cost stabilizing at about 4 seconds. This is due to the fact that the exploration area for the Viola & Jones object detection models for both the faces and the eyes have a bigger area to explore, but the radial symmetry algorithm still maintains the same cost, working over a bounding box region that is rescaled to a fixed size. Past the $8.5 \cdot 10^5$ pixels mark, the algorithm costs stabilizes because of a downscaling of the image to a maximum width of 1024 pixels.

Figures 37a and 37b show the mean temporal cost of processing an image relative to the *minimum face heterogeneity* and *test radii range* parameter values. For the *minimum face heterogeneity* parameter, it can be seen that while the temporal cost remains constant until a value of 0.06 is reached, past which it descends linearly. This is due to the fact that past this value face candidates are being discarded because of a low face heterogeneity, which prevents the algorithm from further analysing them.

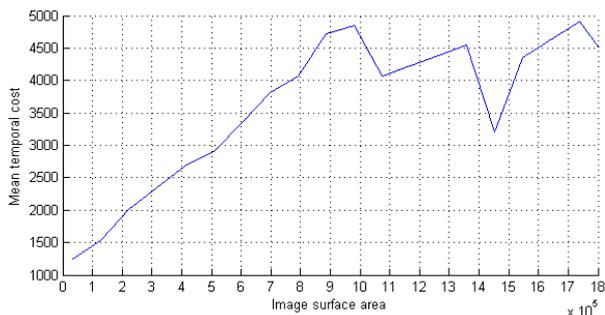
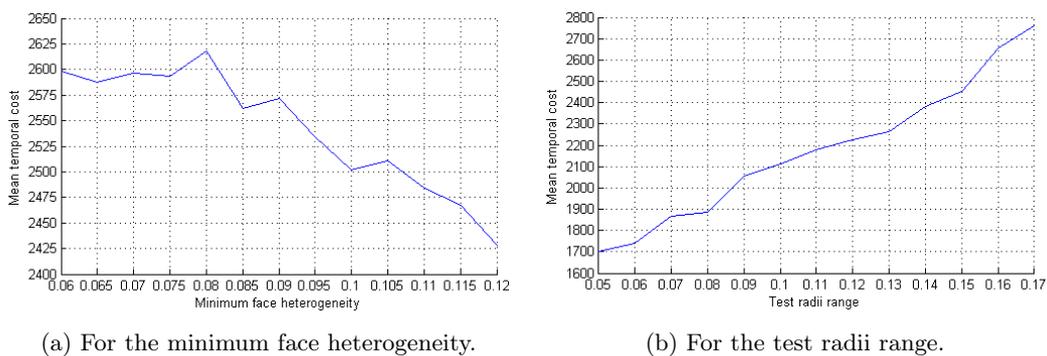


Figure 36: Mean temporal cost of processing an image relative to the surface area of the image.



(a) For the minimum face heterogeneity.

(b) For the test radii range.

Figure 37: Mean temporal cost of the methodology relative to the value of the method parameters affecting the number of operations to perform.

In the case of the *test radii range*, which represents the difference between the maximum and minimum radial symmetry test radii, the cost increases linearly. This result was to expect, since the eye centre detection is the most costly step of the algorithm, and the number of times the fast radial symmetry transform technique must be applied is directly proportional to the test radii range.

When comparing the increases in the mean time taken to process an image in function of the three analysed parameters, it can be seen that the most significant one is the image size. This is due to the algorithm having a bigger area to perform face detection, and in consequence a higher probability of detecting false positives, which must be analysed afterwards before discarding them.

7 Conclusions

In this work, a methodology has been proposed for face detection and eyes centre location in still images based on the *Viola & Jones object detection* and the *fast radial symmetry transform* techniques. The method uses specific information on the problem domain in order to increase the face detection rate and eyes centre location precision without increasing the False Positive rate. In order to do so, the mentioned techniques have been combined with domain-specific knowledge to filter out False Positives and increase the eyes centre location precision, allowing the initial detection process to be relaxed. Also, the domain-specific knowledge has been used to model multivariate normal distributions that allow for the face and eye centre probabilities to be estimated.

The proposed method has turned out to be highly accurate both in terms of face detection and eyes centre location, with a relevant statistical estimation of the detection probabilities for the

faces that is capable of assessing the validity of the detections under certain circumstances. This permits the discarding of invalid faces when more than one is detected in a controlled environment, which is the purpose of the statistic. As such, it has been shown that the probability of a face being correctly detected can be estimated from the relation of the Viola & Jones bounding boxes for the face itself and the eyes bounding box, these maintaining a specific relative ratio and position that is characteristic of the trained detectors and that follows a multivariate normal distribution in the case of faces.

Also, the usage of a set of constraints in order to filter the face candidates by using the configuration of the face and eyes bounding boxes has been proven successful in discarding the False Positives of the face, with only 38 False Positives detected over a set of 1000 images.

With a 96.1% mean accuracy over face detection, the algorithm has also been found to have a variable accuracy with relation to gender and ethnicity, with an accuracy over the 97% in the case of Europeans, Asians and South Americans, and low accuracies for Middle Eastern (95.5%) and African (90%) people. In the case of eyes centre location, the methodology has been found to have a high precision, detecting more than 90% of the cases with an error margin lower than the 3% of face width, and all of the eye centres of within a 13% of the face width for the detected faces.

In terms of temporal cost, the algorithm has been shown to vary linearly with relation to the image surface area and the minimum face heterogeneity and range of test radii parameters. Since the two methodology parameters are fixed when deploying the library, and considering the parameters selected by using the whole 1000-images dataset as the final ones, it can be seen that the algorithm is expected to take between 1.25 and 5 seconds to process an image, with the upper bound being at 5 seconds, since the image size is rescaled past a certain size. Also, it has been found that the most influential parameter is the image size, not the algorithm parameters themselves. This is due to a bigger area to analyse which implies a higher cost for the Viola & Jones face detection algorithm, as well as a higher chance of detecting false positives, which must be analysed before discarding them.

When compared to the solution developed by *Neurotechnologies*, the proposed methodology has been shown to have a slightly better, although not statistically significant, face detection accuracy relative to the allowed error margin. For the eyes centre detection, on the other hand, the accuracy is much higher and statistically significant for small error margins, which means the eye centres are located with higher precision.

8 Future work

In this work a methodology for the detection of faces and location of the eye centres in individual images has been developed, implemented and evaluated which uses the context information of the images in order to increase the accuracy of more generic methods. This concept could be extended to the temporal domain, using the context information of video sequences to locate the desired features more precisely on individual frames.

More precisely, it is possible to create probabilistic models which take into account the detected features as well as the probabilities assigned to them at each frame to help determine the location of these features on contiguous frames. Two techniques to consider are *Markov chains* and *Kalman filters*, with both approaches using the information on the detections at the frame level over the whole sequence in order to correct detection errors at individual frames.

List of Figures

1	Sample images processed by the methodology	5
2	General flow of a facial expression analysis application	6
3	Hierarchy of facial feature extraction and representation methods	7
4	Hierarchy of facial expression recognition methods	9
5	Hierarchy of multi-modal emotion recognition fusion methods	10
6	General steps of the methodology.	11
7	Haar-like features set example	12
8	Fast radial symmetry transform algorithm voting regions	12
9	Pseudo-code for the face and eye false positives discarding methodology	14
10	Pseudocode for the eyes centre location methodology	16
11	Q-Q plot for the eyes mean surface and the standard normal distributions	17
12	Q-Q plot for the intra-ocular distance and the standard normal distributions	18
13	Q-Q plot for the eyes mean height and the standard normal distribution	18
14	Q-Q plot for the eyes pair vertical difference and the standard normal distributions	20
15	Q-Q plot for the eyes pair horizontal difference and the standard normal distributions	20
16	Bounds of the face framing to be detected by the algorithm.	21
17	Use cases diagram for the developed library.	22
18	Classes diagram for the developed library.	23
19	Sequence diagram describing the library initialization process.	29
20	Sequence diagram describing the <i>parameters definition</i> use case.	30
21	Sequence diagram describing the <i>face and eye centres detection</i> use case.	31
22	Sequence diagram describing the <i>results elimination</i> use case.	32
23	Sequence diagram describing the <i>error message obtention</i> use case.	32
24	Sample images from the dataset	33
25	Partitions selection for each 10-fold cross-validation fold	34
26	FP and TP rates relative to the minimum face heterogeneity parameter	36
27	FP and TP rates relative to the maximum eyes cross-correlation parameter	37
28	FP and TP rates relative to the maximum and minimum eyes distance parameters	38
29	True Positive rates relative to the minimum and maximum test radii parameters	39
30	AUC for the error margin vs. True Positives rate plot relative to the eyes binary mask threshold	40

31	AUC for the error margin vs. True Positives rate plot relative to the fast radial symmetry degree	41
32	Recall relative to error margin for face detection and eyes centre location	43
33	Sample processed images from the dataset	45
34	ROC curve for face detection	46
35	Accuracies comparison between the <i>Faceanalyser</i> and <i>Neurotechnologies</i> solutions .	47
36	Temporal cost relative to the input image surface area	48
37	Temporal cost relative to the relevant methodology parameters	48

List of Tables

1	Means vector for the faces multivariate normal distribution.	19
2	Covariance matrix for the faces multivariate normal distribution.	19
3	Means vector for the eyes multivariate normal distribution.	20
4	Covariance matrix for the eyes multivariate normal distribution.	21
5	Minimum face heterogeneity 10-fold cross-validation parameter selection	36
6	Minimum eyes pair cross-correlation 10-fold cross-validation parameter selection	37
7	Intra-ocular distance 10-fold cross-validation parameter selection	38
8	Radial symmetry test radii 10-fold cross-validation parameter selection	39
9	Eyes binary mask threshold 10-fold cross-validation parameter selection	40
10	Fast radial symmetry degree 10-fold cross-validation parameter selection	41
11	Parameters selected at each fold of the 10-fold cross-validation	42
12	10-fold cross-validation test accuracies and overall methodology accuracy	42
13	Test accuracies by ethnicity and gender	43
14	Selected parameter values to perform method comparison	46

A Accuracy tables for the face heterogeneity parameter selection

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.040	0.9611 ± 0.0064	0.9644 ± 0.0059	0.9589 ± 0.0066	0.9622 ± 0.0064	0.9589 ± 0.0066
0.045	0.9611 ± 0.0064	0.9644 ± 0.0062	0.9589 ± 0.0066	0.9622 ± 0.0064	0.9589 ± 0.0066
0.050	0.9611 ± 0.0064	0.9644 ± 0.0062	0.9589 ± 0.0066	0.9622 ± 0.0064	0.9589 ± 0.0066
0.055	0.9611 ± 0.0064	0.9644 ± 0.0062	0.9589 ± 0.0066	0.9622 ± 0.0064	0.9589 ± 0.0066
0.060	0.9611 ± 0.0064	0.9644 ± 0.0062	0.9589 ± 0.0066	0.9622 ± 0.0064	0.9589 ± 0.0066
0.065	0.9611 ± 0.0064	0.9644 ± 0.0062	0.9589 ± 0.0066	0.9622 ± 0.0064	0.9589 ± 0.0066
0.070	0.9600 ± 0.0065	0.9622 ± 0.0064	0.9567 ± 0.0068	0.9600 ± 0.0065	0.9578 ± 0.0067
0.075	0.9589 ± 0.0066	0.9600 ± 0.0065	0.9544 ± 0.0070	0.9578 ± 0.0067	0.9556 ± 0.0069
0.080	0.9578 ± 0.0067	0.9589 ± 0.0066	0.9533 ± 0.0070	0.9567 ± 0.0068	0.9544 ± 0.0070
0.085	0.9489 ± 0.0073	0.9489 ± 0.0073	0.9433 ± 0.0077	0.9478 ± 0.0074	0.9444 ± 0.0076
0.090	0.9444 ± 0.0076	0.9433 ± 0.0077	0.9378 ± 0.0081	0.9433 ± 0.0077	0.9400 ± 0.0079
0.095	0.9322 ± 0.0084	0.9289 ± 0.0086	0.9256 ± 0.0087	0.9289 ± 0.0086	0.9278 ± 0.0086
0.100	0.9211 ± 0.0090	0.9189 ± 0.0091	0.9122 ± 0.0094	0.9167 ± 0.0092	0.9156 ± 0.0093
0.105	0.9089 ± 0.0096	0.9078 ± 0.0096	0.9011 ± 0.0100	0.9056 ± 0.0097	0.9067 ± 0.0097
0.110	0.8833 ± 0.0107	0.8833 ± 0.0107	0.8789 ± 0.0109	0.8822 ± 0.0107	0.8822 ± 0.0107
0.115	0.8667 ± 0.0113	0.8656 ± 0.0114	0.8667 ± 0.0113	0.8656 ± 0.0114	0.8656 ± 0.0114
0.120	0.8278 ± 0.0126	0.8289 ± 0.0126	0.8333 ± 0.0124	0.8278 ± 0.0126	0.8289 ± 0.0126

	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
0.040	0.9600 ± 0.0065	0.9567 ± 0.0068	0.9600 ± 0.0065	0.9578 ± 0.0067	0.9589 ± 0.0066
0.045	0.9600 ± 0.0065	0.9567 ± 0.0068	0.9600 ± 0.0065	0.9578 ± 0.0067	0.9589 ± 0.0066
0.050	0.9600 ± 0.0065	0.9567 ± 0.0068	0.9600 ± 0.0065	0.9578 ± 0.0067	0.9589 ± 0.0066
0.055	0.9600 ± 0.0065	0.9567 ± 0.0068	0.9600 ± 0.0065	0.9578 ± 0.0067	0.9589 ± 0.0066
0.060	0.9600 ± 0.0065	0.9567 ± 0.0068	0.9600 ± 0.0065	0.9578 ± 0.0067	0.9589 ± 0.0066
0.065	0.9600 ± 0.0065	0.9567 ± 0.0068	0.9600 ± 0.0065	0.9578 ± 0.0067	0.9589 ± 0.0066
0.070	0.9578 ± 0.0067	0.9544 ± 0.0070	0.9578 ± 0.0067	0.9556 ± 0.0069	0.9567 ± 0.0068
0.075	0.9556 ± 0.0069	0.9522 ± 0.0071	0.9556 ± 0.0069	0.9544 ± 0.0070	0.9544 ± 0.0070
0.080	0.9544 ± 0.0070	0.9511 ± 0.0072	0.9544 ± 0.0070	0.9533 ± 0.0070	0.9544 ± 0.0070
0.085	0.9478 ± 0.0074	0.9422 ± 0.0078	0.9456 ± 0.0076	0.9444 ± 0.0076	0.9456 ± 0.0076
0.090	0.9433 ± 0.0077	0.9367 ± 0.0081	0.9411 ± 0.0078	0.9389 ± 0.0080	0.9400 ± 0.0079
0.095	0.9300 ± 0.0085	0.9244 ± 0.0088	0.9289 ± 0.0086	0.9256 ± 0.0087	0.9267 ± 0.0087
0.100	0.9167 ± 0.0092	0.9144 ± 0.0093	0.9156 ± 0.0093	0.9144 ± 0.0093	0.9133 ± 0.0094
0.105	0.9078 ± 0.0096	0.9022 ± 0.0099	0.9033 ± 0.0099	0.9033 ± 0.0099	0.9022 ± 0.0099
0.110	0.8856 ± 0.0106	0.8756 ± 0.0110	0.8811 ± 0.0108	0.8789 ± 0.0109	0.8778 ± 0.0109
0.115	0.8678 ± 0.0113	0.8622 ± 0.0115	0.8656 ± 0.0114	0.8611 ± 0.0115	0.8622 ± 0.0115
0.120	0.8356 ± 0.0124	0.8244 ± 0.0127	0.8311 ± 0.0125	0.8244 ± 0.0127	0.8267 ± 0.0126

B Accuracy tables for the eyes heterogeneity parameter selection

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.080	0.9667 ± 0.0060	0.9689 ± 0.0058	0.9633 ± 0.0063	0.9667 ± 0.0060	0.9644 ± 0.0062
0.090	0.9667 ± 0.0060	0.9689 ± 0.0058	0.9633 ± 0.0063	0.9667 ± 0.0060	0.9644 ± 0.0062
0.100	0.9667 ± 0.0060	0.9689 ± 0.0058	0.9633 ± 0.0063	0.9667 ± 0.0060	0.9644 ± 0.0062
0.110	0.9678 ± 0.0059	0.9700 ± 0.0057	0.9644 ± 0.0062	0.9678 ± 0.0059	0.9656 ± 0.0061
0.120	0.9678 ± 0.0059	0.9700 ± 0.0057	0.9644 ± 0.0062	0.9678 ± 0.0059	0.9656 ± 0.0061
0.130	0.9678 ± 0.0059	0.9700 ± 0.0057	0.9644 ± 0.0062	0.9678 ± 0.0059	0.9656 ± 0.0061
0.140	0.9678 ± 0.0059	0.9700 ± 0.0057	0.9644 ± 0.0062	0.9678 ± 0.0059	0.9656 ± 0.0061
0.150	0.9678 ± 0.0059	0.9700 ± 0.0057	0.9644 ± 0.0062	0.9678 ± 0.0059	0.9656 ± 0.0061
0.160	0.9678 ± 0.0059	0.9700 ± 0.0057	0.9644 ± 0.0062	0.9678 ± 0.0059	0.9656 ± 0.0061
0.170	0.9678 ± 0.0059	0.9711 ± 0.0056	0.9656 ± 0.0061	0.9689 ± 0.0058	0.9667 ± 0.0060
0.180	0.9678 ± 0.0059	0.9711 ± 0.0056	0.9656 ± 0.0061	0.9689 ± 0.0058	0.9667 ± 0.0060
0.190	0.9678 ± 0.0059	0.9711 ± 0.0056	0.9656 ± 0.0061	0.9689 ± 0.0058	0.9667 ± 0.0060
0.200	0.9678 ± 0.0059	0.9711 ± 0.0056	0.9656 ± 0.0061	0.9689 ± 0.0058	0.9667 ± 0.0060
0.210	0.9678 ± 0.0059	0.9711 ± 0.0056	0.9656 ± 0.0061	0.9689 ± 0.0058	0.9667 ± 0.0060
0.220	0.9678 ± 0.0059	0.9711 ± 0.0056	0.9656 ± 0.0061	0.9689 ± 0.0058	0.9667 ± 0.0060

	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
0.080	0.9656 ± 0.0061	0.9622 ± 0.0064	0.9644 ± 0.0062	0.9633 ± 0.0063	0.9644 ± 0.0062
0.090	0.9656 ± 0.0061	0.9622 ± 0.0064	0.9644 ± 0.0062	0.9633 ± 0.0063	0.9644 ± 0.0062
0.100	0.9656 ± 0.0061	0.9622 ± 0.0064	0.9644 ± 0.0062	0.9633 ± 0.0063	0.9644 ± 0.0062
0.110	0.9667 ± 0.0060	0.9633 ± 0.0063	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9656 ± 0.0061
0.120	0.9667 ± 0.0060	0.9633 ± 0.0063	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9656 ± 0.0061
0.130	0.9667 ± 0.0060	0.9633 ± 0.0063	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9656 ± 0.0061
0.140	0.9667 ± 0.0060	0.9633 ± 0.0063	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9656 ± 0.0061
0.150	0.9667 ± 0.0060	0.9633 ± 0.0063	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9656 ± 0.0061
0.160	0.9667 ± 0.0060	0.9633 ± 0.0063	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9656 ± 0.0061
0.170	0.9678 ± 0.0059	0.9644 ± 0.0062	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9667 ± 0.0060
0.180	0.9678 ± 0.0059	0.9644 ± 0.0062	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9667 ± 0.0060
0.190	0.9678 ± 0.0059	0.9644 ± 0.0062	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9667 ± 0.0060
0.200	0.9678 ± 0.0059	0.9644 ± 0.0062	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9667 ± 0.0060
0.210	0.9678 ± 0.0059	0.9644 ± 0.0062	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9667 ± 0.0060
0.220	0.9678 ± 0.0059	0.9644 ± 0.0062	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9667 ± 0.0060

C Accuracy tables for the eyes min/max distance parameters selection

Fold 1					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9633 ± 0.0063	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9656 ± 0.0061
0.290	0.9633 ± 0.0063	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9656 ± 0.0061
0.300	0.9622 ± 0.0064	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9644 ± 0.0062
0.310	0.9611 ± 0.0064	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063
0.320	0.9556 ± 0.0069	0.9578 ± 0.0067	0.9578 ± 0.0067	0.9578 ± 0.0067	0.9578 ± 0.0067

Fold 2					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9656 ± 0.0061	0.9678 ± 0.0059	0.9678 ± 0.0059	0.9678 ± 0.0059	0.9678 ± 0.0059
0.290	0.9656 ± 0.0061	0.9678 ± 0.0059	0.9678 ± 0.0059	0.9678 ± 0.0059	0.9678 ± 0.0059
0.300	0.9644 ± 0.0062	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060
0.310	0.9644 ± 0.0062	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060
0.320	0.9589 ± 0.0066	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064

Fold 3					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9600 ± 0.0065	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064
0.290	0.9600 ± 0.0065	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064
0.300	0.9589 ± 0.0066	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064
0.310	0.9578 ± 0.0067	0.9600 ± 0.0065	0.9600 ± 0.0065	0.9600 ± 0.0065	0.9600 ± 0.0065
0.320	0.9544 ± 0.0070	0.9567 ± 0.0068	0.9567 ± 0.0068	0.9567 ± 0.0068	0.9567 ± 0.0068

Fold 4					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9644 ± 0.0062	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060
0.290	0.9644 ± 0.0062	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060
0.300	0.9644 ± 0.0062	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060	0.9667 ± 0.0060
0.310	0.9633 ± 0.0063	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9656 ± 0.0061	0.9656 ± 0.0061
0.320	0.9600 ± 0.0065	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064

Fold 5					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9611 ± 0.0064	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063
0.290	0.9611 ± 0.0064	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063
0.300	0.9600 ± 0.0065	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064
0.310	0.9589 ± 0.0066	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064
0.320	0.9533 ± 0.0070	0.9556 ± 0.0069	0.9556 ± 0.0069	0.9556 ± 0.0069	0.9556 ± 0.0069

Fold 6					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9622 ± 0.0064	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9644 ± 0.0062
0.290	0.9622 ± 0.0064	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9644 ± 0.0062	0.9644 ± 0.0062
0.300	0.9611 ± 0.0064	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063
0.310	0.9600 ± 0.0065	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064
0.320	0.9544 ± 0.0070	0.9567 ± 0.0068	0.9567 ± 0.0068	0.9567 ± 0.0068	0.9567 ± 0.0068

Fold 7					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9600 ± 0.0065	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064
0.290	0.9600 ± 0.0065	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064
0.300	0.9589 ± 0.0066	0.9600 ± 0.0065	0.9600 ± 0.0065	0.9600 ± 0.0065	0.9600 ± 0.0065
0.310	0.9578 ± 0.0067	0.9589 ± 0.0066	0.9589 ± 0.0066	0.9589 ± 0.0066	0.9589 ± 0.0066
0.320	0.9522 ± 0.0071	0.9533 ± 0.0070	0.9533 ± 0.0070	0.9533 ± 0.0070	0.9533 ± 0.0070

Fold 8					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9611 ± 0.0064	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063
0.290	0.9611 ± 0.0064	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063
0.300	0.9600 ± 0.0065	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064
0.310	0.9589 ± 0.0066	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064
0.320	0.9533 ± 0.0070	0.9556 ± 0.0069	0.9556 ± 0.0069	0.9556 ± 0.0069	0.9556 ± 0.0069

Fold 9					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9611 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064
0.290	0.9611 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064
0.300	0.9600 ± 0.0065	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064
0.310	0.9589 ± 0.0066	0.9600 ± 0.0065	0.9600 ± 0.0065	0.9600 ± 0.0065	0.9600 ± 0.0065
0.320	0.9533 ± 0.0070	0.9544 ± 0.0070	0.9544 ± 0.0070	0.9544 ± 0.0070	0.9544 ± 0.0070

Fold 10					
	0.480	0.490	0.500	0.510	0.520
0.280	0.9611 ± 0.0064	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063
0.290	0.9611 ± 0.0064	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063	0.9633 ± 0.0063
0.300	0.9600 ± 0.0065	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064	0.9622 ± 0.0064
0.310	0.9589 ± 0.0066	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064	0.9611 ± 0.0064
0.320	0.9544 ± 0.0070	0.9567 ± 0.0068	0.9567 ± 0.0068	0.9567 ± 0.0068	0.9567 ± 0.0068

D Accuracy tables for the eyes symmetry min/max test radii parameters selection

Fold 1				
	0.160	0.170	0.180	0.190
0.080	0.97255 ± 0.00092	0.97414 ± 0.00087	0.97479 ± 0.00085	0.97511 ± 0.00084
0.090	0.97266 ± 0.00091	0.97432 ± 0.00086	0.97489 ± 0.00084	0.97526 ± 0.00083
0.100	0.97294 ± 0.00090	0.97459 ± 0.00085	0.97522 ± 0.00083	0.97555 ± 0.00082
0.110	0.97321 ± 0.00089	0.97468 ± 0.00085	0.97535 ± 0.00082	0.97579 ± 0.00081
0.120	0.97391 ± 0.00085	0.97516 ± 0.00082	0.97565 ± 0.00081	0.97589 ± 0.00080
0.130	0.97403 ± 0.00085	0.97512 ± 0.00083	0.97535 ± 0.00082	0.97567 ± 0.00081
0.140	0.97399 ± 0.00086	0.97499 ± 0.00084	0.97508 ± 0.00083	0.97530 ± 0.00084
0.150	0.97452 ± 0.00085	0.97472 ± 0.00085	0.97493 ± 0.00084	0.97518 ± 0.00084

Fold 1				
	0.200	0.210	0.220	0.230
0.080	0.97547 ± 0.00083	0.97582 ± 0.00082	0.97575 ± 0.00082	0.97585 ± 0.00081
0.090	0.97564 ± 0.00082	0.97596 ± 0.00081	0.97588 ± 0.00081	0.97600 ± 0.00080
0.100	0.97579 ± 0.00081	0.97606 ± 0.00080	0.97598 ± 0.00080	0.97609 ± 0.00079
0.110	0.97591 ± 0.00081	0.97615 ± 0.00080	0.97605 ± 0.00080	0.97621 ± 0.00079
0.120	0.97616 ± 0.00080	0.97628 ± 0.00079	0.97608 ± 0.00079	0.97622 ± 0.00078
0.130	0.97590 ± 0.00080	0.97589 ± 0.00080	0.97572 ± 0.00080	0.97584 ± 0.00079
0.140	0.97556 ± 0.00082	0.97567 ± 0.00081	0.97544 ± 0.00081	0.97555 ± 0.00080
0.150	0.97542 ± 0.00082	0.97547 ± 0.00081	0.97529 ± 0.00080	0.97534 ± 0.00079

Fold 2				
	0.160	0.170	0.180	0.190
0.080	0.97269 ± 0.00091	0.97418 ± 0.00087	0.97481 ± 0.00086	0.97533 ± 0.00084
0.090	0.97287 ± 0.00091	0.97439 ± 0.00087	0.97494 ± 0.00085	0.97545 ± 0.00084
0.100	0.97315 ± 0.00089	0.97470 ± 0.00085	0.97524 ± 0.00084	0.97583 ± 0.00082
0.110	0.97353 ± 0.00088	0.97480 ± 0.00085	0.97549 ± 0.00082	0.97589 ± 0.00082
0.120	0.97403 ± 0.00086	0.97522 ± 0.00084	0.97560 ± 0.00082	0.97595 ± 0.00081
0.130	0.97427 ± 0.00085	0.97538 ± 0.00082	0.97539 ± 0.00083	0.97586 ± 0.00082
0.140	0.97454 ± 0.00084	0.97524 ± 0.00083	0.97534 ± 0.00084	0.97565 ± 0.00083
0.150	0.97492 ± 0.00084	0.97504 ± 0.00085	0.97498 ± 0.00085	0.97536 ± 0.00084

Fold 2				
	0.200	0.210	0.220	0.230
0.080	0.97559 ± 0.00083	0.97589 ± 0.00083	0.97599 ± 0.00082	0.97598 ± 0.00081
0.090	0.97576 ± 0.00082	0.97603 ± 0.00082	0.97613 ± 0.00081	0.97613 ± 0.00080
0.100	0.97592 ± 0.00082	0.97624 ± 0.00080	0.97623 ± 0.00080	0.97622 ± 0.00080
0.110	0.97594 ± 0.00081	0.97614 ± 0.00081	0.97611 ± 0.00081	0.97616 ± 0.00080
0.120	0.97600 ± 0.00082	0.97607 ± 0.00081	0.97599 ± 0.00081	0.97602 ± 0.00081
0.130	0.97589 ± 0.00082	0.97598 ± 0.00081	0.97582 ± 0.00081	0.97582 ± 0.00081
0.140	0.97565 ± 0.00083	0.97580 ± 0.00082	0.97556 ± 0.00083	0.97554 ± 0.00082
0.150	0.97548 ± 0.00083	0.97540 ± 0.00083	0.97530 ± 0.00082	0.97526 ± 0.00081

Fold 3				
	0.160	0.170	0.180	0.190
0.080	0.97258 ± 0.00093	0.97417 ± 0.00088	0.97481 ± 0.00086	0.97522 ± 0.00085
0.090	0.97279 ± 0.00092	0.97439 ± 0.00087	0.97495 ± 0.00085	0.97540 ± 0.00084
0.100	0.97306 ± 0.00091	0.97471 ± 0.00085	0.97532 ± 0.00084	0.97572 ± 0.00082
0.110	0.97346 ± 0.00089	0.97488 ± 0.00085	0.97559 ± 0.00082	0.97592 ± 0.00082
0.120	0.97422 ± 0.00085	0.97533 ± 0.00083	0.97579 ± 0.00082	0.97602 ± 0.00081
0.130	0.97428 ± 0.00085	0.97539 ± 0.00082	0.97544 ± 0.00083	0.97576 ± 0.00082
0.140	0.97450 ± 0.00085	0.97522 ± 0.00084	0.97524 ± 0.00084	0.97546 ± 0.00084
0.150	0.97474 ± 0.00085	0.97499 ± 0.00085	0.97516 ± 0.00085	0.97542 ± 0.00084

Fold 3				
	0.200	0.210	0.220	0.230
0.080	0.97562 ± 0.00083	0.97583 ± 0.00083	0.97589 ± 0.00082	0.97592 ± 0.00082
0.090	0.97581 ± 0.00082	0.97599 ± 0.00082	0.97604 ± 0.00081	0.97608 ± 0.00081
0.100	0.97600 ± 0.00082	0.97624 ± 0.00080	0.97619 ± 0.00081	0.97622 ± 0.00080
0.110	0.97615 ± 0.00081	0.97628 ± 0.00080	0.97621 ± 0.00081	0.97630 ± 0.00080
0.120	0.97624 ± 0.00081	0.97630 ± 0.00080	0.97615 ± 0.00080	0.97622 ± 0.00080
0.130	0.97593 ± 0.00081	0.97597 ± 0.00080	0.97575 ± 0.00082	0.97581 ± 0.00081
0.140	0.97574 ± 0.00083	0.97580 ± 0.00081	0.97548 ± 0.00083	0.97552 ± 0.00082
0.150	0.97565 ± 0.00082	0.97550 ± 0.00082	0.97534 ± 0.00082	0.97536 ± 0.00080

Fold 4				
	0.160	0.170	0.180	0.190
0.080	0.97245 ± 0.00093	0.97389 ± 0.00088	0.97454 ± 0.00085	0.97492 ± 0.00084
0.090	0.97255 ± 0.00092	0.97400 ± 0.00087	0.97462 ± 0.00085	0.97504 ± 0.00084
0.100	0.97275 ± 0.00091	0.97427 ± 0.00086	0.97494 ± 0.00084	0.97532 ± 0.00083
0.110	0.97327 ± 0.00089	0.97457 ± 0.00085	0.97534 ± 0.00082	0.97560 ± 0.00082
0.120	0.97397 ± 0.00085	0.97516 ± 0.00083	0.97556 ± 0.00081	0.97582 ± 0.00081
0.130	0.97416 ± 0.00085	0.97523 ± 0.00081	0.97531 ± 0.00082	0.97566 ± 0.00081
0.140	0.97445 ± 0.00084	0.97511 ± 0.00083	0.97518 ± 0.00083	0.97536 ± 0.00083
0.150	0.97468 ± 0.00084	0.97487 ± 0.00084	0.97501 ± 0.00083	0.97520 ± 0.00083

Fold 4				
	0.200	0.210	0.220	0.230
0.080	0.97524 ± 0.00084	0.97548 ± 0.00083	0.97559 ± 0.00082	0.97559 ± 0.00082
0.090	0.97538 ± 0.00083	0.97562 ± 0.00082	0.97572 ± 0.00081	0.97573 ± 0.00081
0.100	0.97553 ± 0.00082	0.97582 ± 0.00081	0.97581 ± 0.00081	0.97582 ± 0.00081
0.110	0.97578 ± 0.00081	0.97591 ± 0.00080	0.97588 ± 0.00081	0.97590 ± 0.00080
0.120	0.97605 ± 0.00080	0.97601 ± 0.00080	0.97590 ± 0.00080	0.97590 ± 0.00080
0.130	0.97585 ± 0.00080	0.97574 ± 0.00080	0.97557 ± 0.00081	0.97552 ± 0.00081
0.140	0.97558 ± 0.00082	0.97552 ± 0.00081	0.97529 ± 0.00082	0.97526 ± 0.00082
0.150	0.97548 ± 0.00081	0.97525 ± 0.00082	0.97508 ± 0.00081	0.97506 ± 0.00080

Fold 5				
	0.160	0.170	0.180	0.190
0.080	0.97251 ± 0.00093	0.97407 ± 0.00087	0.97487 ± 0.00085	0.97537 ± 0.00083
0.090	0.97265 ± 0.00092	0.97424 ± 0.00087	0.97497 ± 0.00084	0.97551 ± 0.00083
0.100	0.97292 ± 0.00091	0.97454 ± 0.00085	0.97532 ± 0.00083	0.97582 ± 0.00081
0.110	0.97331 ± 0.00089	0.97469 ± 0.00085	0.97557 ± 0.00081	0.97597 ± 0.00081
0.120	0.97405 ± 0.00085	0.97520 ± 0.00083	0.97576 ± 0.00081	0.97606 ± 0.00080
0.130	0.97404 ± 0.00086	0.97512 ± 0.00082	0.97540 ± 0.00082	0.97578 ± 0.00081
0.140	0.97439 ± 0.00085	0.97507 ± 0.00084	0.97527 ± 0.00083	0.97546 ± 0.00083
0.150	0.97452 ± 0.00086	0.97485 ± 0.00085	0.97502 ± 0.00084	0.97528 ± 0.00084

Fold 5				
	0.200	0.210	0.220	0.230
0.080	0.97576 ± 0.00082	0.97615 ± 0.00081	0.97615 ± 0.00081	0.97619 ± 0.00080
0.090	0.97592 ± 0.00082	0.97629 ± 0.00080	0.97629 ± 0.00080	0.97633 ± 0.00079
0.100	0.97610 ± 0.00081	0.97651 ± 0.00079	0.97640 ± 0.00079	0.97644 ± 0.00079
0.110	0.97622 ± 0.00080	0.97651 ± 0.00079	0.97638 ± 0.00079	0.97648 ± 0.00078
0.120	0.97634 ± 0.00079	0.97648 ± 0.00079	0.97631 ± 0.00079	0.97640 ± 0.00078
0.130	0.97605 ± 0.00080	0.97622 ± 0.00079	0.97599 ± 0.00080	0.97607 ± 0.00079
0.140	0.97576 ± 0.00082	0.97599 ± 0.00080	0.97574 ± 0.00081	0.97582 ± 0.00080
0.150	0.97560 ± 0.00082	0.97563 ± 0.00082	0.97544 ± 0.00082	0.97548 ± 0.00080

Fold 6				
	0.160	0.170	0.180	0.190
0.080	0.97255 ± 0.00093	0.97404 ± 0.00088	0.97483 ± 0.00085	0.97531 ± 0.00084
0.090	0.97269 ± 0.00093	0.97422 ± 0.00087	0.97493 ± 0.00085	0.97545 ± 0.00084
0.100	0.97302 ± 0.00091	0.97459 ± 0.00086	0.97535 ± 0.00083	0.97577 ± 0.00082
0.110	0.97353 ± 0.00088	0.97494 ± 0.00085	0.97573 ± 0.00081	0.97605 ± 0.00081
0.120	0.97411 ± 0.00086	0.97527 ± 0.00083	0.97582 ± 0.00081	0.97604 ± 0.00081
0.130	0.97425 ± 0.00086	0.97533 ± 0.00083	0.97550 ± 0.00083	0.97574 ± 0.00082
0.140	0.97450 ± 0.00085	0.97515 ± 0.00084	0.97524 ± 0.00084	0.97556 ± 0.00084
0.150	0.97471 ± 0.00086	0.97499 ± 0.00085	0.97513 ± 0.00085	0.97545 ± 0.00083

Fold 6				
	0.200	0.210	0.220	0.230
0.080	0.97569 ± 0.00083	0.97603 ± 0.00082	0.97600 ± 0.00082	0.97597 ± 0.00081
0.090	0.97585 ± 0.00082	0.97616 ± 0.00081	0.97613 ± 0.00081	0.97611 ± 0.00080
0.100	0.97603 ± 0.00081	0.97639 ± 0.00080	0.97625 ± 0.00080	0.97623 ± 0.00080
0.110	0.97628 ± 0.00080	0.97651 ± 0.00079	0.97636 ± 0.00080	0.97639 ± 0.00079
0.120	0.97635 ± 0.00080	0.97652 ± 0.00080	0.97631 ± 0.00080	0.97630 ± 0.00079
0.130	0.97602 ± 0.00081	0.97616 ± 0.00080	0.97589 ± 0.00081	0.97592 ± 0.00081
0.140	0.97592 ± 0.00082	0.97598 ± 0.00081	0.97568 ± 0.00082	0.97568 ± 0.00082
0.150	0.97581 ± 0.00082	0.97569 ± 0.00082	0.97559 ± 0.00081	0.97556 ± 0.00080

Fold 7				
	0.160	0.170	0.180	0.190
0.080	0.97229 ± 0.00094	0.97378 ± 0.00089	0.97442 ± 0.00087	0.97501 ± 0.00085
0.090	0.97248 ± 0.00093	0.97400 ± 0.00088	0.97457 ± 0.00087	0.97519 ± 0.00085
0.100	0.97277 ± 0.00092	0.97433 ± 0.00087	0.97496 ± 0.00085	0.97555 ± 0.00083
0.110	0.97315 ± 0.00090	0.97455 ± 0.00087	0.97527 ± 0.00083	0.97572 ± 0.00083
0.120	0.97371 ± 0.00087	0.97491 ± 0.00085	0.97535 ± 0.00083	0.97570 ± 0.00082
0.130	0.97384 ± 0.00087	0.97500 ± 0.00084	0.97510 ± 0.00084	0.97555 ± 0.00083
0.140	0.97408 ± 0.00086	0.97492 ± 0.00085	0.97500 ± 0.00085	0.97531 ± 0.00085
0.150	0.97440 ± 0.00086	0.97477 ± 0.00086	0.97485 ± 0.00086	0.97518 ± 0.00084

Fold 7				
	0.200	0.210	0.220	0.230
0.080	0.97546 ± 0.00084	0.97581 ± 0.00083	0.97584 ± 0.00082	0.97580 ± 0.00082
0.090	0.97565 ± 0.00083	0.97598 ± 0.00082	0.97600 ± 0.00081	0.97598 ± 0.00081
0.100	0.97594 ± 0.00082	0.97630 ± 0.00080	0.97622 ± 0.00081	0.97619 ± 0.00080
0.110	0.97607 ± 0.00081	0.97632 ± 0.00080	0.97623 ± 0.00081	0.97626 ± 0.00080
0.120	0.97611 ± 0.00081	0.97628 ± 0.00081	0.97611 ± 0.00081	0.97612 ± 0.00080
0.130	0.97590 ± 0.00082	0.97603 ± 0.00080	0.97578 ± 0.00081	0.97578 ± 0.00081
0.140	0.97558 ± 0.00083	0.97573 ± 0.00081	0.97543 ± 0.00083	0.97542 ± 0.00082
0.150	0.97561 ± 0.00082	0.97559 ± 0.00082	0.97544 ± 0.00081	0.97541 ± 0.00080

Fold 8				
	0.160	0.170	0.180	0.190
0.080	0.97214 ± 0.00094	0.97372 ± 0.00089	0.97455 ± 0.00087	0.97501 ± 0.00085
0.090	0.97226 ± 0.00094	0.97388 ± 0.00089	0.97462 ± 0.00086	0.97510 ± 0.00085
0.100	0.97250 ± 0.00093	0.97417 ± 0.00087	0.97495 ± 0.00085	0.97542 ± 0.00084
0.110	0.97290 ± 0.00091	0.97433 ± 0.00087	0.97522 ± 0.00083	0.97557 ± 0.00083
0.120	0.97382 ± 0.00087	0.97500 ± 0.00084	0.97554 ± 0.00082	0.97579 ± 0.00082
0.130	0.97385 ± 0.00087	0.97496 ± 0.00084	0.97511 ± 0.00084	0.97542 ± 0.00083
0.140	0.97407 ± 0.00086	0.97479 ± 0.00085	0.97490 ± 0.00085	0.97509 ± 0.00085
0.150	0.97446 ± 0.00086	0.97447 ± 0.00087	0.97465 ± 0.00086	0.97501 ± 0.00085

Fold 8				
	0.200	0.210	0.220	0.230
0.080	0.97532 ± 0.00085	0.97561 ± 0.00084	0.97558 ± 0.00083	0.97555 ± 0.00083
0.090	0.97546 ± 0.00084	0.97570 ± 0.00083	0.97567 ± 0.00083	0.97564 ± 0.00082
0.100	0.97566 ± 0.00083	0.97594 ± 0.00081	0.97580 ± 0.00082	0.97577 ± 0.00082
0.110	0.97581 ± 0.00082	0.97597 ± 0.00081	0.97582 ± 0.00082	0.97584 ± 0.00082
0.120	0.97607 ± 0.00081	0.97614 ± 0.00081	0.97591 ± 0.00081	0.97591 ± 0.00081
0.130	0.97569 ± 0.00082	0.97573 ± 0.00081	0.97543 ± 0.00083	0.97541 ± 0.00082
0.140	0.97536 ± 0.00084	0.97548 ± 0.00083	0.97516 ± 0.00084	0.97509 ± 0.00083
0.150	0.97531 ± 0.00083	0.97514 ± 0.00084	0.97499 ± 0.00083	0.97496 ± 0.00082

Fold 9				
	0.160	0.170	0.180	0.190
0.080	0.97270 ± 0.00093	0.97421 ± 0.00088	0.97476 ± 0.00086	0.97514 ± 0.00085
0.090	0.97277 ± 0.00093	0.97430 ± 0.00088	0.97482 ± 0.00086	0.97526 ± 0.00085
0.100	0.97301 ± 0.00092	0.97454 ± 0.00086	0.97513 ± 0.00085	0.97555 ± 0.00084
0.110	0.97328 ± 0.00090	0.97469 ± 0.00086	0.97547 ± 0.00083	0.97575 ± 0.00083
0.120	0.97402 ± 0.00087	0.97520 ± 0.00084	0.97571 ± 0.00082	0.97588 ± 0.00082
0.130	0.97418 ± 0.00086	0.97528 ± 0.00083	0.97539 ± 0.00083	0.97568 ± 0.00083
0.140	0.97422 ± 0.00086	0.97501 ± 0.00085	0.97505 ± 0.00085	0.97534 ± 0.00085
0.150	0.97447 ± 0.00087	0.97476 ± 0.00086	0.97489 ± 0.00086	0.97523 ± 0.00085

Fold 9				
	0.200	0.210	0.220	0.230
0.080	0.97543 ± 0.00084	0.97574 ± 0.00084	0.97573 ± 0.00083	0.97572 ± 0.00083
0.090	0.97556 ± 0.00084	0.97586 ± 0.00083	0.97584 ± 0.00082	0.97584 ± 0.00082
0.100	0.97572 ± 0.00083	0.97611 ± 0.00081	0.97599 ± 0.00082	0.97598 ± 0.00081
0.110	0.97596 ± 0.00082	0.97626 ± 0.00081	0.97612 ± 0.00082	0.97613 ± 0.00081
0.120	0.97611 ± 0.00082	0.97629 ± 0.00081	0.97606 ± 0.00081	0.97609 ± 0.00081
0.130	0.97591 ± 0.00082	0.97607 ± 0.00081	0.97578 ± 0.00082	0.97582 ± 0.00081
0.140	0.97562 ± 0.00084	0.97583 ± 0.00082	0.97549 ± 0.00083	0.97551 ± 0.00082
0.150	0.97550 ± 0.00083	0.97540 ± 0.00083	0.97521 ± 0.00083	0.97522 ± 0.00081

Fold 10				
	0.160	0.170	0.180	0.190
0.080	0.97218 ± 0.00094	0.97364 ± 0.00089	0.97450 ± 0.00087	0.97496 ± 0.00085
0.090	0.97224 ± 0.00094	0.97374 ± 0.00089	0.97453 ± 0.00086	0.97506 ± 0.00085
0.100	0.97241 ± 0.00093	0.97401 ± 0.00088	0.97482 ± 0.00086	0.97527 ± 0.00084
0.110	0.97295 ± 0.00091	0.97431 ± 0.00087	0.97509 ± 0.00084	0.97546 ± 0.00084
0.120	0.97376 ± 0.00087	0.97488 ± 0.00085	0.97535 ± 0.00083	0.97561 ± 0.00083
0.130	0.97380 ± 0.00087	0.97488 ± 0.00084	0.97497 ± 0.00085	0.97532 ± 0.00084
0.140	0.97402 ± 0.00087	0.97466 ± 0.00086	0.97474 ± 0.00086	0.97498 ± 0.00086
0.150	0.97427 ± 0.00087	0.97456 ± 0.00087	0.97473 ± 0.00086	0.97497 ± 0.00086

Fold 10				
	0.200	0.210	0.220	0.230
0.080	0.97536 ± 0.00084	0.97560 ± 0.00084	0.97560 ± 0.00083	0.97558 ± 0.00083
0.090	0.97540 ± 0.00084	0.97563 ± 0.00084	0.97563 ± 0.00083	0.97562 ± 0.00083
0.100	0.97562 ± 0.00083	0.97590 ± 0.00082	0.97579 ± 0.00083	0.97577 ± 0.00082
0.110	0.97577 ± 0.00082	0.97592 ± 0.00082	0.97582 ± 0.00082	0.97584 ± 0.00082
0.120	0.97594 ± 0.00082	0.97600 ± 0.00082	0.97581 ± 0.00082	0.97582 ± 0.00081
0.130	0.97562 ± 0.00083	0.97573 ± 0.00082	0.97551 ± 0.00083	0.97550 ± 0.00082
0.140	0.97530 ± 0.00084	0.97546 ± 0.00083	0.97519 ± 0.00084	0.97516 ± 0.00083
0.150	0.97533 ± 0.00084	0.97525 ± 0.00084	0.97510 ± 0.00083	0.97504 ± 0.00082

E Accuracy table for the eyes binary mask threshold parameter selection

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.160	0.9755 ± 0.0008	0.9755 ± 0.0008	0.9756 ± 0.0008	0.9752 ± 0.0008	0.9753 ± 0.0008
0.180	0.9755 ± 0.0008	0.9754 ± 0.0008	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9754 ± 0.0008
0.200	0.9757 ± 0.0008	0.9755 ± 0.0008	0.9757 ± 0.0008	0.9754 ± 0.0008	0.9756 ± 0.0008
0.220	0.9759 ± 0.0008	0.9757 ± 0.0008	0.9759 ± 0.0008	0.9756 ± 0.0008	0.9756 ± 0.0008
0.240	0.9759 ± 0.0008	0.9758 ± 0.0008	0.9759 ± 0.0008	0.9757 ± 0.0008	0.9757 ± 0.0008
0.260	0.9761 ± 0.0008	0.9759 ± 0.0008	0.9760 ± 0.0008	0.9758 ± 0.0008	0.9757 ± 0.0008
0.280	0.9761 ± 0.0008	0.9759 ± 0.0008	0.9760 ± 0.0008	0.9757 ± 0.0008	0.9758 ± 0.0008
0.300	0.9758 ± 0.0008	0.9758 ± 0.0008	0.9759 ± 0.0008	0.9756 ± 0.0008	0.9757 ± 0.0008
0.320	0.9757 ± 0.0008	0.9757 ± 0.0008	0.9758 ± 0.0008	0.9755 ± 0.0008	0.9756 ± 0.0008

	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
0.160	0.9754 ± 0.0008	0.9751 ± 0.0008	0.9753 ± 0.0008	0.9754 ± 0.0008	0.9749 ± 0.0008
0.180	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9754 ± 0.0008	0.9754 ± 0.0008	0.9750 ± 0.0008
0.200	0.9756 ± 0.0008	0.9753 ± 0.0008	0.9755 ± 0.0008	0.9757 ± 0.0008	0.9751 ± 0.0008
0.220	0.9757 ± 0.0008	0.9755 ± 0.0008	0.9757 ± 0.0008	0.9759 ± 0.0008	0.9754 ± 0.0008
0.240	0.9758 ± 0.0008	0.9756 ± 0.0008	0.9758 ± 0.0008	0.9759 ± 0.0008	0.9754 ± 0.0008
0.260	0.9760 ± 0.0008	0.9757 ± 0.0008	0.9758 ± 0.0008	0.9759 ± 0.0008	0.9755 ± 0.0008
0.280	0.9760 ± 0.0008	0.9756 ± 0.0008	0.9758 ± 0.0008	0.9759 ± 0.0008	0.9755 ± 0.0008
0.300	0.9759 ± 0.0008	0.9755 ± 0.0008	0.9756 ± 0.0008	0.9758 ± 0.0008	0.9754 ± 0.0008
0.320	0.9757 ± 0.0008	0.9754 ± 0.0008	0.9754 ± 0.0008	0.9757 ± 0.0008	0.9753 ± 0.0008

F Accuracy table for the fast radial symmetry degree parameter selection

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.750	0.9755 ± 0.0008	0.9754 ± 0.0008	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008
1.000	0.9766 ± 0.0008	0.9764 ± 0.0008	0.9768 ± 0.0008	0.9766 ± 0.0008	0.9766 ± 0.0008
1.250	0.9766 ± 0.0008	0.9764 ± 0.0008	0.9768 ± 0.0008	0.9766 ± 0.0008	0.9766 ± 0.0008
1.500	0.9766 ± 0.0008	0.9764 ± 0.0008	0.9768 ± 0.0008	0.9766 ± 0.0008	0.9766 ± 0.0008
1.750	0.9766 ± 0.0008	0.9764 ± 0.0008	0.9768 ± 0.0008	0.9766 ± 0.0008	0.9766 ± 0.0008
2.000	0.9755 ± 0.0008	0.9754 ± 0.0008	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008
2.250	0.9755 ± 0.0008	0.9754 ± 0.0008	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008
2.500	0.9755 ± 0.0008	0.9754 ± 0.0008	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008
2.750	0.9755 ± 0.0008	0.9754 ± 0.0008	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008
3.000	0.9735 ± 0.0009	0.9736 ± 0.0009	0.9737 ± 0.0009	0.9734 ± 0.0009	0.9735 ± 0.0009

	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
0.750	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008	0.9751 ± 0.0008
1.000	0.9766 ± 0.0008	0.9764 ± 0.0008	0.9762 ± 0.0008	0.9766 ± 0.0008	0.9764 ± 0.0008
1.250	0.9766 ± 0.0008	0.9764 ± 0.0008	0.9762 ± 0.0008	0.9766 ± 0.0008	0.9764 ± 0.0008
1.500	0.9766 ± 0.0008	0.9764 ± 0.0008	0.9762 ± 0.0008	0.9766 ± 0.0008	0.9764 ± 0.0008
1.750	0.9766 ± 0.0008	0.9764 ± 0.0008	0.9762 ± 0.0008	0.9766 ± 0.0008	0.9764 ± 0.0008
2.000	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008	0.9751 ± 0.0008
2.250	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008	0.9751 ± 0.0008
2.500	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008	0.9751 ± 0.0008
2.750	0.9755 ± 0.0008	0.9752 ± 0.0008	0.9752 ± 0.0008	0.9755 ± 0.0008	0.9751 ± 0.0008
3.000	0.9737 ± 0.0009	0.9733 ± 0.0009	0.9732 ± 0.0009	0.9737 ± 0.0009	0.9731 ± 0.0009

References

- [1] J. Luetten B. Fasela. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36:259–275, 2003.
- [2] R. Hammoud L. Trujillo E. Romero B. Hernandez, G. Olague. Visual learning of texture descriptors for facial expression recognition in thermal imagery. *Computer Vision and Image Understanding*, 106:258–269, 2007.
- [3] M. Lang B. Schuller, G. Rigoll. Hidden markov model-based speech emotion recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2, 2003.
- [4] Y.M. Huang C.L. Huang. Facial expression recognition using model-based feature extraction and action parameters classification. *Journal of Visual Communication and Image Representation*, 8:278–290, 1997.
- [5] S. Purushothaman D. Arumugam. Emotion classification using facial expression. *International Journal of Advanced Computer Science and Applications*, 2, 2011.
- [6] G. Volpe N. Dael K. Scherer D. Glowinski, A. Camurri. Technique for automatic emotion recognition by body gesture analysis. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, 2008.
- [7] F. Girosit E. Osuna, R. Freund. Training support vector machines: an application to face detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 130–136, 1997.
- [8] Matteo Ferrara, Annalisa Franco, and Davide Maltoni. Evaluating systems assessing face-image compliance with icao/iso standards. In *Biometrics and Identity Management*, pages 191–199. Springer, 2008.
- [9] L. Kessous A. Raouzaoui L. Malatesta S. Asteriadis K. Karpouzis G. Caridakis, G. Castellano. Multimodal emotion recognition from expressive faces, body gestures and speech. *Artificial intelligence and innovations 2007: From theory to applications*, pages 375–388, 2007.
- [10] A. Zelinsky G. Loy. A fast radial symmetry transform for detecting points of interest. *European Conference on Computer Vision*, pages 358–368, 2002.
- [11] T. Jan H. Gunes, M. Piccardi. Bimodal modelling of facial and upper-body gesture for affective hci. *Annual Conference of the Human Factors and Ergonomics Society*, 2004.
- [12] H. Demirel H. Soyel. Facial expression recognition using 3d facial feature distances. *Image Analysis and Recognition*, pages 831–838, 2007.
- [13] T.S. Huang H. Tang. 3d facial expression recognition based on automatically selected features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.
- [14] T. Kanade H.A. Rowley, S. Baluja. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:23–38, 1998.
- [15] T. Kanade H.A. Rowley, S. Baluja. Rotation invariant neural network-based face detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 20:38–44, 1998.
- [16] L.X. Zhen J.C. Huang H.B. Deng, L.W. Jin. A new facial expression recognition method based on local gabor filter bank and pca plus lda. *International Journal of Information Technology*, 11:86–96, 2005.
- [17] William G Hunter and J Stuart Hunter. Statistics for experimenters. *Interscience*, New York, page 130, 1978.

- [18] I. Pitas I. Kotsia. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Transactions on Image Processing*, 16:172–187, 2007.
- [19] F. Lingenfelter J. Kim. Ensemble approaches to parametric decision fusion for bimodal emotion recognition. *Biosignals*, pages 460–463, 2010.
- [20] E. Andre J. Wagner, J. Kim. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. *IEEE International Conference on Multimedia and Expo*, pages 940–943, 2005.
- [21] X. Wei Y. Sun J. Wang, L. Yin. 3d facial expression recognition based on primitive surface feature distribution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:1399–1406, 2006.
- [22] C.W. Omlin J. Whitehill. Haar features for faces au recognition. *7th International Conference on Automatic Face and Gesture Recognition*, 2006.
- [23] S.R. Kim K.H. Kim, S.W. Bang. Emotion recognition system using short-term monitoring of physiological signals. *Medical and biological engineering and computing*, 42:419–427, 2004.
- [24] J. Kim. Bimodal emotion recognition using speech and physiological changes. *Robust speech recognition and understanding*, pages 265–280, 2007.
- [25] T. Poggio K.K. Sung. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:39–51, 1998.
- [26] R. Hammoud B. Hernandez L. Trujillo, G. Olague. Automatic feature localization in thermal images for facial expression recognition. *Computer Society Conference on Computer Vision and Pattern Recognition*, pages 14–14, 2005.
- [27] Y. Sun T. Worm M. Reale L. Yin, X. Chen. A high-resolution 3d dynamic facial expression database. *8th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008.
- [28] R. Nakatsu L.C. De Silva, T. Miyasato. Facial emotion recognition using multimodal information. *Proceedings of International Conference on Information, Communications and Signal Processing*, pages 397–401, 1997.
- [29] T.S. Hai L.H. Thai, N.D.T. Nguyen. A facial expression classification system integrating canny, principal component analysis and artificial neural network. *International Journal of Machine Learning and Computing*, 1:388–393, 2011.
- [30] T. Miyasato R. Nakatsu L.S. Chen, T.S. Huang. Multimodal human emotion / expression recognition. *Proceedings of International Conference on Automatic Face and Gesture Recognition*, 1998.
- [31] T.S. Huang L.S. Chen. Emotional expressions in audiovisual human computer interaction. *IEEE International Conference on Multimedia*, 1:423–426, 2000.
- [32] P. Kuchi M. Gargesha. Facial expression recognition using artificial neural networks. *Artificial Neural Computation Systems*, pages 1–6, 2002.
- [33] M. Kamachi J. Gyoba M. Lyons, S. Akamatsu. Coding facial expressions with gabor wavelets. *Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998.
- [34] I. Fasel J.R. Movellan M.S. Bartlett, G. Littlewort. Real time face detection and facial expression recognition: Development and applications to human computer interaction. *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 5:53–53, 2003.
- [35] M. Frank C. Lainscsek I. Fasel J. Movellan M.S. Bartlett, G. Littlewort. Recognizing facial expression: Machine learning and application to spontaneous behavior. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

- [36] M.G. Frank C. Lainscsek I.R. Fasel J.R. Movellan M.S. Bartlett, G.C. Littlewort. Automatic recognition of facial actions in spontaneous expressions. *Journal of Multimedia*, 1:22–35, 2006.
- [37] I. Cohen T. Gevers T.S. Huang N. Sebe, E. Bakker. Bimodal emotion recognition. *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, 2005.
- [38] M. Jones P. Viola. Rapid object detection using a boosted cascade of simple features. *Conference on Computer Vision and Pattern Recognition*, 2001.
- [39] M.J. Jones P. Viola. Robust real-time face detection. *International journal of computer vision*, 57:137–154, 2004.
- [40] Q. Ji P. Wang. Multi-view face detection under complex scene based on combined svms. *IEEE International Conference on Pattern Recognition*, pages 179–182, 2004.
- [41] W.V.Friesen P.Ekman. Facial action coding system. *Consulting Pshychologists Press Inc.*, 1978.
- [42] A.K. Katsaggelos P.S. Aleksic. Automatic facial expression recognition using facial animation parameters and multistream hmms. *IEEE Transactions on Information Forensics and Security*, 1:3–11, 2006.
- [43] A. Apatean S. Emerich, E. Lupu. Analysis of emotion recognition using facial expressions, speech and multimodal information. *17th European Signal Processing Conference*, 2009.
- [44] P. Sharma. Feature based method for human facial emotion detection using optical flow based analysis. *International Journal of Research in Computer Science*, 1:31–38, 2011.
- [45] G. Bakiri T.G. Dietterich. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
- [46] L.C. De Silva T.L. Nwe, S.W. Foo. Speech emotion recognition using hidden markov models. *Speech communication*, 41:603–623, 2003.
- [47] J.F. Cohn Y. Tian, T. Kanade. Facial expression analysis. *Handbook of face recognition*, 2005.
- [48] Q. Ji Y. Tong, W. Liao. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
- [49] B. Wu C. Huang Y. Wang, H. Ai. Real time facial expression recognition with adaboost. *Proceedings of the 17th International Conference on Pattern Recognition*, 3:926–929, 2004.
- [50] T. Kawano T. Kilazoe Y. Yoshitomi, S.I. Kim. Effect of sensor fusion for recognition of emotional states using voice, face image and thermal image of face. *9th IEEE International Workshop on Robot and Human Interactive Communication*, pages 178–183, 2000.